

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関  
国際事務局

(43) 国際公開日  
2012 年 9 月 27 日(27.09.2012)



(10) 国際公開番号  
**WO 2012/128120 A1**

- (51) 国際特許分類:  
**G06F 11/36** (2006.01)
- (21) 国際出願番号: PCT/JP2012/056368
- (22) 国際出願日: 2012 年 3 月 13 日(13.03.2012)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:  
特願 2011-060199 2011 年 3 月 18 日(18.03.2011) JP
- (71) 出願人 (米国を除く全ての指定国について): 独立行政法人産業技術総合研究所(NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY) [JP/JP]; 〒1008921 東京都千代田区霞が関 1-3-1 Tokyo (JP).
- (72) 発明者: および
- (75) 発明者/出願人 (米国についてのみ): 安藤 崇央 (ANDO, Takahiro) [JP/JP]; 〒6610974 兵庫県尼崎市若王寺 3-11-46 独立行政法人産業技術総合研究所尼崎支所内 Hyogo (JP). 崔 銀恵 (CHOI, Eun hye) [KR/JP]; 〒6610974 兵庫県尼崎市若王寺 3-11-46 独立行政法人産業技術総合研究所尼崎支所内 Hyogo (JP). 西原 秀明 (NISHIHARA, Hideaki) [JP/JP]; 〒6610974 兵庫県尼

崎市若王寺 3-11-46 独立行政法人産業技術総合研究所尼崎支所内 Hyogo (JP).

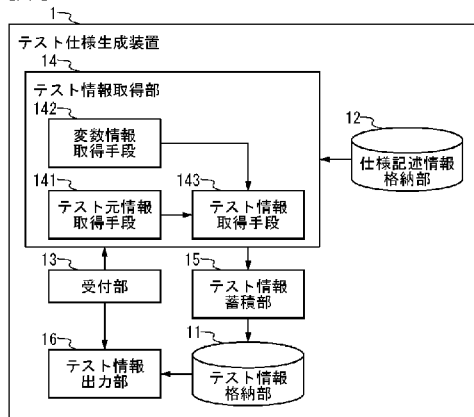
- (74) 代理人: 谷川 英和(TANIGAWA, Hidekazu); 〒5400008 大阪府大阪市中央区大手前 1 丁目 7-31 OMMビル 8 階 私書箱 53 号 Osaka (JP).
- (81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR),

[続葉有]

(54) Title: TEST SPECIFICATION GENERATOR, TEST SPECIFICATION GENERATION METHOD, AND PROGRAM

(54) 発明の名称: テスト仕様生成装置、テスト仕様生成方法、およびプログラム

[図1]



- 1 Test specification generator  
11 Test information storage unit  
12 Specification description information storage unit  
13 Receiver  
14 Test information acquisition unit  
15 Test information accumulation unit  
16 Test information output unit  
141 Test source information acquisition means  
142 Variable information acquisition means  
143 Test information acquisition means

(57) Abstract: Provided is a test specification generator comprising: a test information storage unit which can store test information that is information used in a test; a specification description information storage unit which can store specification description information that is information related to a target specification described in a specification description language constituting a language for describing the target specification, that is information including at least one variable declaration, and that is information including test source information for generating the test information; a test information acquisition unit that uses the variable declaration and test source information to acquire at least one item of the test information having at least the input value of at least one variable; and a test information accumulation unit for accumulating, in the test information storage unit, at least one item of the test information acquired by the test information acquisition unit. Thus, if information for test design is described at the time of the target design, information related to the appropriate test specification can be automatically generated.

(57) 要約: テストで利用する情報であるテスト情報を格納し得るテスト情報格納部と、対象の仕様について記述する言語である仕様記述言語で記述された対象の仕様に関する情報であり、1 以上の変数の宣言を含む情報であり、テスト情報を生成するためのテスト元情報を含む情報である仕様記述情報を格納し得る仕様記述情報格納部と、変数の宣言およびテスト元情報を用いて、1 以上の各変数の入力値を少なくとも有する 1 以上のテスト情報を取得するテスト情報取得部と、テスト情報取得部が取得した 1 以上のテスト情報をテスト情報格納部に蓄積するテスト情報蓄積部とを具備するテスト仕様

生成装置により、対象の設計時にテスト設計のための情報を記載すれば、適切なテスト仕様に関する情報が自動生成できる。

**WO 2012/128120 A1**



---

OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:

— 国際調査報告 (条約第 21 条(3))

## 明 細 書

発明の名称：

テスト仕様生成装置、テスト仕様生成方法、およびプログラム

### 技術分野

[0001] 本発明は、仕様記述言語で記述された仕様記述情報から、テスト仕様を生成するテスト仕様生成装置等に関するものである。

### 背景技術

[0002] 従来、UMLのような自然文を含む仕様記述言語による仕様記述において、自然言語の特性から来る作成効率、品質の問題を改善する仕様記述支援方法があった（例えば、特許文献1参照）。かかる仕様記述支援方法において、変更可能部分と変更不能部分とで構成される自然文のテンプレートからひとつを選択し、テンプレートの変更可能部分に用語の入力を行う。また、本仕様記述支援方法において、用語の辞書を有し、未登録の用語について、用語の登録を促すことを特徴とする。

### 先行技術文献

#### 特許文献

[0003] 特許文献1：特開2008-15879号公報（第1頁、第1図等）

### 発明の概要

#### 発明が解決しようとする課題

[0004] しかしながら、従来技術において、設計段階での仕様記述の支援は行えても、テスト工程の支援は不十分で、テスト設計のために多大な労力を要していた。

#### 課題を解決するための手段

[0005] 本第一の発明のテスト仕様生成装置は、テストで利用する情報であるテスト情報を格納し得るテスト情報格納部と、対象の仕様について記述する言語である仕様記述言語で記述された対象の仕様に関する情報であり、1以上の

変数の宣言を含む情報であり、テスト情報を生成するためのテスト元情報を含む情報である仕様記述情報を格納し得る仕様記述情報格納部と、変数の宣言およびテスト元情報を用いて、1以上の各変数の入力値を少なくとも有する1以上のテスト情報を取得するテスト情報取得部と、テスト情報取得部が取得した1以上のテスト情報をテスト情報格納部に蓄積するテスト情報蓄積部とを具備するテスト仕様生成装置である。

[0006] かかる構成により、対象の設計時に仕様に関する情報を形式的に記載すれば、適切なテスト仕様に関する情報が自動生成できる。

[0007] また、本第二の発明のテスト仕様生成装置は、第一の発明に対して、変数は、型を有し、仕様記述情報は、1以上の変数のうち、1以上の各変数に対応付けて、各変数が採り得る値の中で、テストで使用する値を特定する情報であるテスト元情報を有し、テスト情報取得部は、仕様記述情報から、テスト元情報を変数と対応付けて取得するテスト元情報取得手段と、仕様記述情報から、テスト元情報を有さない変数と変数の型を取得する変数情報取得手段と、テスト元情報取得手段が取得したテスト元情報から、各テスト元情報に対応する変数の1以上の入力値を変数ごとに取得し、かつ、変数情報取得手段が取得した変数と変数の型から入力値を変数ごとに取得するテスト情報取得手段とを具備するテスト仕様生成装置である。

[0008] かかる構成により、対象の設計時に、形式的に記載された仕様の中で、変数に対応付けてテストで使用する値を特定すれば、適切なテスト仕様に関する情報が自動生成できる。

[0009] また、本第三の発明のテスト仕様生成装置は、第二の発明に対して、テスト情報取得手段は、テスト元情報取得手段が取得したテスト元情報から、各テスト元情報に対応する変数の1以上の入力値および出力値を変数ごとに取得し、かつ、変数情報取得手段が取得した変数と変数の型から入力値および出力値を変数ごとに取得するテスト仕様生成装置である。

[0010] かかる構成により、対象の設計時に、形式的に記載された仕様の中で、変数に対応付けてテストで使用する値を特定すれば、適切なテストスイートが

自動生成できる。

[0011] また、本第四の発明のテスト仕様生成装置は、第三の発明に対して、仕様記述情報は、1 以上の変数の宣言、および制約に関する記述である 1 以上の制約情報を有し、テスト情報取得手段は、テスト元情報取得手段が取得したテスト元情報から、各テスト元情報に対応する変数の 1 以上の入力値および出力値を変数ごとに取得し、かつ、変数情報取得手段が取得した変数と変数の型から入力値および出力値を変数ごとに取得し、かつ、すべての制約情報を満たすように、1 以上の入力値および 1 以上の出力値の組であるテストスイートを 1 以上取得するテスト仕様生成装置である。

[0012] かかる構成により、対象の設計時に、形式的に記載された仕様の中で、変数に対応付けてテストで使用する値を特定すれば、適切なテストスイートが自動生成できる。

[0013] また、本第五の発明のテスト仕様生成装置は、第四の発明に対して、制約情報は、1 以上の変数の値の集合を有する状態である遷移前状態と、遷移前状態に対して発生するイベントと、遷移前状態の場合にイベントが発生した場合に移行する状態である遷移後状態とを有する状態遷移情報、または 1 以上の変数の値の集合を有する第一状態と第一状態の場合に必ず採用される 1 以上の変数の値の集合を有する第二状態とを有する不変条件情報であるテスト仕様生成装置である。

[0014] かかる構成により、対象の設計時に、形式的に記載された仕様の中で、変数に対応付けてテストで使用する値を特定すれば、適切なテストスイートが自動生成できる。

[0015] また、本第六の発明のテスト仕様生成装置は、第五の発明に対して、テスト情報取得手段は、1 以上の制約情報が有する遷移前状態とイベントとの組である 1 以上の前提情報をも制約情報として用いて、すべての制約情報を満たすような、1 以上の入力値および 1 以上の出力値の組であるテストスイートを 1 以上取得するテスト仕様生成装置である。

[0016] かかる構成により、状態遷移の前提条件を満たさないテストスイートの生

成を防止できる。

[0017] また、本第七の発明のテスト仕様生成装置は、第五の発明に対して、テスト情報取得部は、イベントに関する制約を示す１以上のイベント制約情報を格納し得るイベント制約情報格納手段をさらに具備し、テスト情報取得手段は、１以上のイベント制約情報をも制約情報として用いて、すべての制約情報を満たすように、１以上の入力値および１以上の出力値の組であるテストスイートを１以上取得するテスト仕様生成装置である。

[0018] かかる構成により、指定したイベントを含むテストスイートの生成を防止できる。

[0019] また、本第八の発明のテスト仕様生成装置は、第五の発明に対して、テスト情報取得手段は、既に取得したテストスイートに対応する情報から、遷移前状態の情報を削除し、かつ、遷移前状態以外の情報が有する部分の論理否定を構成した１以上の第二制約情報を取得し、１以上の第二制約情報をも制約情報として用いて、すべての制約情報を満たすように、１以上の入力値および１以上の出力値の組であるテストスイートを１以上取得するテスト仕様生成装置である。

[0020] かかる構成により、遷移後状態として既知のものを持つテストスイートの生成を防止できる。

[0021] また、本第九の発明のテスト仕様生成装置は、第五の発明に対して、テスト情報取得手段は、既に取得したテストスイートに対応する情報から、遷移後状態の情報を削除し、かつ、遷移後状態以外の情報が有する部分の論理否定を構成した１以上の第二制約情報を取得し、１以上の第二制約情報をも制約情報として用いて、すべての制約情報を満たすように、１以上の入力値および１以上の出力値の組であるテストスイートを１以上取得するテスト仕様生成装置である。

[0022] かかる構成により、遷移前状態として既知のものを持つテストスイートの生成を防止できる。

## 発明の効果

[0023] 本発明によるテスト仕様生成装置によれば、対象の設計時に、形式的に記載された仕様の中で、テスト設計のための情報を記載すれば、適切なテスト仕様に関する情報が自動生成できる。

### 図面の簡単な説明

- [0024] [図1]実施の形態1におけるテスト仕様生成装置のブロック図  
[図2]同テスト仕様生成装置の動作について説明するフローチャート  
[図3]同変数表取得処理について説明するフローチャート  
[図4]同命題論理式取得処理について説明するフローチャート  
[図5]同テスト情報取得処理について説明するフローチャート  
[図6]同仕様記述情報を示す図  
[図7]同変数表を示す図  
[図8]同命題論理式の集合の例を示す図  
[図9]同テスト情報取得手段が取得した数値列を示す図  
[図10]同テスト情報を示す図  
[図11]同テスト情報取得部の動作の概念について説明する図  
[図12]同テスト仕様生成装置の処理の流れを説明する図  
[図13]実施の形態2におけるテスト仕様生成装置2のブロック図  
[図14]同仕様記述情報を示す図  
[図15]同変数表を示す図  
[図16]同命題論理式の集合の例を示す図  
[図17]同テスト情報取得手段が取得した数値列を示す図  
[図18]同無用なテストスイートの抑制を図る動作を説明する図  
[図19]実施の形態1におけるコンピュータシステムの概観図  
[図20]同コンピュータシステムのブロック図

### 発明を実施するための形態

[0025] 以下、テスト仕様生成装置等の実施形態について図面を参照して説明する。なお、実施の形態において同じ符号を付した構成要素は同様の動作を行うので、再度の説明を省略する場合がある。

## [0026] (実施の形態 1)

本実施の形態において、仕様記述言語で記述された仕様記述情報から、テスト仕様を生成する生成テスト仕様生成装置について説明する。なお、仕様記述言語は、通常、形式的な仕様記述言語であるが、その形式を問わない。仕様記述言語は、通常、ソフトウェアを設計・構築する際に、その仕様を記述するために用いられる言語であるが、ハードウェア記述言語でも良い。また、仕様記述言語で記述された仕様記述情報は、プログラムに自動変換されることは好適であるが、必ずしもプログラムに自動変換される必要はない。

[0027] 図 1 は、本実施の形態におけるテスト仕様生成装置 1 のブロック図である。テスト仕様生成装置 1 は、テスト情報格納部 11、仕様記述情報格納部 12、受付部 13、テスト情報取得部 14、テスト情報蓄積部 15、およびテスト情報出力部 16 を備える。

[0028] また、テスト情報取得部 14 は、テスト元情報取得手段 141、変数情報取得手段 142、およびテスト情報取得手段 143 を備える。

[0029] テスト情報格納部 11 は、テスト情報を格納し得る。テスト情報は、対象のテストで利用する情報である。対象とは、仕様記述の対象であり、通常、ソフトウェアであるが、ハードウェアでも良い。また、テスト情報とは、いわゆる 1 以上のテストケースでも良いし、いわゆる 1 以上のテストスイートでも良い。テストケースは、テストの際に対象に与える入力の情報（例えば、1 以上の入力値）である。テストスイートは、テストの際に対象に与える入力の情報（例えば、1 以上の入力値）と、入力の情報を与えた場合の出力の情報（例えば、1 以上の出力値）の組である。テスト情報は、テストの実施に役立つ情報であれば良い。テスト情報は、自動テストのための情報でも良いし、手動でのテストで利用される情報でも良い。なお、出力の情報とは、対象の動作の結果である。

[0030] テスト情報格納部 11 は、不揮発性の記録媒体が好適であるが、揮発性の記録媒体でも実現可能である。

[0031] 仕様記述情報格納部 12 は、仕様記述情報を格納し得る。仕様記述情報は

、仕様記述言語で記述された対象の仕様に関する情報である。また、仕様記述情報は、通常、1 以上の変数の宣言を含む情報である。また、仕様記述情報は、テスト情報を生成するためのテスト元情報を含む情報であることは好適である。仕様記述情報は、1 以上 ( $n$  個) の変数のうち、1 以上の各変数 ( $m$  個,  $n \geq m$ ) に対応付けて、各変数が採り得る値の中で、テストで使用する変数の値を特定する情報であるテスト元情報を有しても良い。つまり、テスト元情報は、テストで使用する 1 以上の値でも良いし、テストで使用する値の範囲 (例えば、1 ~ 10、など) でも良いし、テストで使用する 1 以上の値でも良いし、テストで使用する値の範囲でも良いし、テストで使用する値の条件 (例えば、偶数、3 文字以内のアルファベットの組み合わせの文字列等) でも良いし、テストで使用する値の条件 (例えば、3 で割り切れる自然数等) でも良い。

[0032] また、仕様記述情報は、1 以上の変数の宣言、および制約に関する記述である 1 以上の制約情報を有することは好適である。制約情報は、例えば、状態遷移情報または不変条件情報であることは好適である。また、状態遷移情報とは、状態遷移に関する情報である。状態遷移情報は、遷移前状態とイベントと遷移後状態とアクションとを有することは好適である。状態遷移情報は、遷移前状態とイベントと遷移後状態とを有する情報であっても良い。遷移前状態は、イベントを受け付ける前の状態である。遷移前状態は、1 以上の変数の値の集合を有する状態である。イベントは、遷移前状態に対して発生する。遷移後状態とは、遷移前状態の際にイベントが発生した場合に移行する状態である。遷移後状態は、1 以上の変数の値の集合を有する状態である。アクションは、明示的に特定のイベントを発生させる。また、不変条件情報は、不変条件を示す。不変条件情報は、第一状態と第二状態とを有する。第一状態は、1 以上の変数の値の集合を有する。第二状態は、第一状態の場合に必ず採用される 1 以上の変数の値の集合を有する。不変条件情報は、第一状態でも第二状態でも変わらず成立する条件を示す。また、ここで、変数は、通常、型を有する。型は、データタイプとも言う。型には、採り得る

値や範囲が決まっている型もある。なお、仕様記述情報は、ファイルであることは好適であるが、そのデータ構造は問わない。

[0033] 仕様記述情報格納部 12 は、不揮発性の記録媒体が好適であるが、揮発性の記録媒体でも実現可能である。

[0034] 仕様記述情報格納部 12 に仕様記述情報が記憶される過程は問わない。例えば、記録媒体を介して仕様記述情報が仕様記述情報格納部 12 で記憶されるようになってよく、通信回線等を介して送信された仕様記述情報が仕様記述情報格納部 12 で記憶されるようになってよく、あるいは、入力デバイスを介して入力された仕様記述情報が仕様記述情報格納部 12 で記憶されるようになってよい。

[0035] 受付部 13 は、指示や情報を受け付ける。指示とは、例えば、テスト情報を生成する指示であるテスト情報生成指示である。また、指示とは、例えば、テスト情報を出力する指示であるテスト情報出力指示である。指示などの入力手段は、キーボードやマウスやメニュー画面によるもの等、何でも良い。受付部 13 は、キーボード等の入力手段のデバイスドライバや、メニュー画面の制御ソフトウェア等で実現され得る。

[0036] テスト情報取得部 14 は、仕様記述情報が有する変数の宣言およびテスト元情報を用いて、1 以上の各変数の入力値を少なくとも有する 1 以上のテスト情報を取得する。入力値とは、対象に与える値である。入力値は、変数の初期値や、設定値などでも良い。ここで、取得するとは、生成することを含む。また、取得するとは、記録媒体から読み出すことでも良い。

[0037] テスト情報取得部 14 を構成するテスト元情報取得手段 141 は、仕様記述情報から、テスト元情報を変数と対応付けて取得する。テスト元情報取得手段 141 は、例えば、仕様記述情報に対して、字句解析および構文解析を行い、テスト元情報を変数と対応付けて取得する。また、テスト元情報取得手段 141 は、例えば、仕様記述情報に対して、予約語（例えば、“Var”、“testedwith” など）のパターンマッチング等を行い、テスト元情報を変数と対応付けて取得しても良い。つまり、テスト元情報取得手段 141 は、仕

様記述情報から、予め決められた文字列（例えば、予約語）を用いて、テスト元情報を変数と対応付けて取得しても良い。

[0038] 変数情報取得手段 1 4 2 は、仕様記述情報から、テスト元情報を有さない変数と当該変数の型を取得する。

[0039] テスト情報取得手段 1 4 3 は、テスト元情報取得手段 1 4 1 が取得したテスト元情報から、各テスト元情報に対応する変数の 1 以上の入力値を変数ごとに取得する。また、テスト情報取得手段 1 4 3 は、変数情報取得手段 1 4 2 が取得した変数と変数の型から入力値を変数ごとに取得する。

[0040] また、テスト情報取得手段 1 4 3 は、テスト元情報取得手段 1 4 1 が取得したテスト元情報から、各テスト元情報に対応する変数の 1 以上の入力値および出力値を変数ごとに取得する。また、テスト情報取得手段 1 4 3 は、変数情報取得手段 1 4 2 が取得した変数と変数の型から入力値および出力値を変数ごとに取得しても良い。

[0041] また、テスト情報取得手段 1 4 3 は、テスト元情報取得手段 1 4 1 が取得したテスト元情報から、各テスト元情報に対応する変数の 1 以上の入力値および出力値を変数ごとに取得し、かつ、すべての制約情報を満たすテストスイートを 1 以上取得する。また、テスト情報取得手段 1 4 3 は、変数情報取得手段 1 4 2 が取得した変数と変数の型から入力値および出力値を変数ごとに取得し、かつ、すべての制約情報を満たすテストスイートを 1 以上取得しても良い。

[0042] なお、テストスイートとは、1 以上の入力値および 1 以上の出力値の組である。かかる場合、テスト情報取得手段 1 4 3 は、例えば、テスト元情報取得手段 1 4 1 が取得したテスト元情報と変数の型から、各変数の入力値の候補値を取得する。そして、テスト情報取得手段 1 4 3 は、1 以上の各変数の入力値（状態遷移前の値）と出力値（状態遷移後の値）のすべての組み合わせを取得する。なお、この組み合わせがテストスイートの候補である。そして、テスト情報取得手段 1 4 3 は、テストスイートの候補を、すべての制約情報に適用し、すべての制約情報を満たすテストスイートを、利用するテス

トスイートとして抽出する。なお、１以上の制約情報が存在する場合に、当該１以上の制約情報に与える変数の値の組が、当該１以上の制約情報を満たすか否かを判断する処理は公知技術である。また、テスト情報取得手段１４３は、１以上のテストスイートの候補を取得してから、１以上の制約情報を１以上のテストスイートに適用して、制約情報を満たさないテストスイートを削除しても良い。また、テスト情報取得手段１４３は、テストスイートを取得する際に１以上の制約情報を適用しても良い。

[0043] また、制約情報とは、状態遷移情報または不変条件情報であることは好適である。状態遷移情報とは、１以上の変数の値の集合を有する状態である遷移前状態と遷移前状態に対して発生するイベントと遷移前状態の場合にイベントが発生した場合に移行する状態である遷移後状態とアクションとを有する情報である。また、不変条件情報は、１以上の変数の値の集合を有する第一状態と第一状態の場合に必ず採用される１以上の変数の値の集合を有する第二状態とを有する情報である。

[0044] テスト情報蓄積部１５は、テスト情報取得部１４が取得した１以上のテスト情報をテスト情報格納部１１に蓄積する。

[0045] テスト情報出力部１６は、テスト情報格納部１１のテスト情報を出力する。ここで、出力とは、ディスプレイへの表示、プロジェクターを用いた投影、プリンタへの印字、外部の装置への送信、記録媒体への蓄積、他の処理装置や他のプログラムなどへの処理結果の引渡しなどを含む概念である。テスト情報出力部１６は、ディスプレイ等の出力デバイスを含むと考えるても含まないと考えるても良い。テスト情報出力部１６は、出力デバイスのドライバーソフトまたは、出力デバイスのドライバーソフトと出力デバイス等で実現され得る。

[0046] テスト情報取得部１４、テスト元情報取得手段１４１、変数情報取得手段１４２、テスト情報取得手段１４３、およびテスト情報蓄積部１５は、通常、ＭＰＵやメモリ等から実現され得る。テスト情報取得部１４等の処理手順は、通常、ソフトウェアで実現され、当該ソフトウェアはＲＯＭ等の記録媒

体に記録されている。但し、ハードウェア（専用回路）で実現しても良い。

[0047] 次に、テスト仕様生成装置 1 の動作について、図 2 のフローチャートを用いて説明する。

[0048] （ステップ S 2 0 1）受付部 1 3 は、テスト情報生成指示を受け付けたか否かを判断する。テスト情報生成指示を受け付ければステップ S 2 0 2 に行き、テスト情報生成指示を受け付けなければステップ S 2 0 7 に行く。

[0049] （ステップ S 2 0 2）テスト情報取得部 1 4 は、仕様記述情報格納部 1 2 から仕様記述情報を読み出す。

[0050] （ステップ S 2 0 3）テスト情報取得部 1 4 は、仕様記述情報を解釈し、変数表を取得する。なお、仕様記述情報の解釈とは、通常、字句解析、構文解析等であり、プログラム言語の処置系（例えば、コンパイラやインタプリタ等）が行う公知の技術である。変数表を取得する変数表取得処理について、図 3 のフローチャートを用いて説明する。

[0051] （ステップ S 2 0 4）テスト情報取得部 1 4 は、仕様記述情報を解釈し、命題論理式を取得する。命題論理式を取得する命題論理式取得処理について、図 4 のフローチャートを用いて説明する。ここでの仕様記述情報の解釈も、通常、字句解析、構文解析等であり、プログラム言語の処置系（例えば、コンパイラやインタプリタ等）が行う公知の技術である。

[0052] （ステップ S 2 0 5）テスト情報取得部 1 4 は、変数表と命題論理式とを用いて、テスト情報を取得する。テスト情報を取得するテスト情報取得処理について、図 5 のフローチャートを用いて説明する。

[0053] （ステップ S 2 0 6）テスト情報蓄積部 1 5 は、ステップ S 2 0 5 で取得されたテスト情報を、テスト情報格納部 1 1 に蓄積する。ステップ S 2 0 1 に戻る。

[0054] （ステップ S 2 0 7）受付部 1 3 は、テスト情報出力指示を受け付けたか否かを判断する。テスト情報出力指示を受け付ければステップ S 2 0 8 に行き、テスト情報出力指示を受け付けなければステップ S 2 0 1 に戻る。

[0055] （ステップ S 2 0 8）テスト情報出力部 1 6 は、テスト情報格納部 1 1 か

らテスト情報を読み出す。

[0056] (ステップS 2 0 9) テスト情報出力部 1 6 は、ステップS 2 0 8 で読み出したテスト情報を出力する。ステップS 2 0 1 に戻る。

[0057] なお、図 2 のフローチャートにおいて、電源オフや処理終了の割り込みにより処理は終了する。

[0058] 次に、ステップS 2 0 3 の変数表取得処理について、図 3 のフローチャートを用いて説明する。

[0059] (ステップS 3 0 1) 変数情報取得手段 1 4 2 は、カウンタ  $i$  に 1 を代入する。

[0060] (ステップS 3 0 2) 変数情報取得手段 1 4 2 は、仕様記述情報から  $i$  番目の処理単位を読み出す。なお、一の処理単位とは、例えば、1 行、一の終端子（例えば、“;” や予約語など）の次の文字列から次の終端子まで、特定の予約語の次の文字列から次の特定の予約語まで、などである。一の処理単位とは、変数の宣言か否かを判断する文字列の単位、または制約か否かを判断する文字列の単位等である。

[0061] (ステップS 3 0 3) 変数情報取得手段 1 4 2 は、ステップS 3 0 3 で  $i$  番目の処理単位を読み出せたか否かを判断する。 $i$  番目の処理単位を読み出せればステップS 3 0 4 に行き、読み出せなければ上位処理にリターンする。

[0062] (ステップS 3 0 4) 変数情報取得手段 1 4 2 は、 $i$  番目の処理単位が変数の宣言であるか否かを判断する。変数の宣言であればステップS 3 0 5 に行き、変数の宣言でなければステップS 3 1 0 に行く。なお、処理単位が変数の宣言であるか否かは、例えば、予約語” Var” を含む行か否か、予約語” Var” に続く非予約語であるか否か、変数のタイプ（予約語）を含む行か否か等により判断する。かかる処理単位が変数の宣言であるか否かを判断する処理も公知技術である。

[0063] (ステップS 3 0 5) 変数情報取得手段 1 4 2 は、 $i$  番目の処理単位から変数の型を取得する。

- [0064] (ステップS 3 0 6) テスト元情報取得手段 1 4 1 は、i 番目の処理単位に、テスト元情報が存在するか否かを判断する。テスト元情報が存在すればステップS 3 0 7に行き、テスト元情報が存在しなければステップS 3 1 1に行く。テスト元情報が存在するか否かは、例えば、テスト元情報を特定する予約語（例えば、" testedwith" ）を含むか否かにより行われる。
- [0065] (ステップS 3 0 7) テスト元情報取得手段 1 4 1 は、i 番目の処理単位からテスト元情報を取得する。なお、テスト元情報取得手段 1 4 1 は、例えば、テスト元情報を特定する予約語（例えば、" testedwith" ）に対応するテスト元情報を取得する。
- [0066] (ステップS 3 0 8) 変数情報取得手段 1 4 2 は、テスト元情報と型、またはテスト元情報を用いて、変数の 1 以上の入力値、または変数の 1 以上の入力値と変数の 1 以上の出力値を取得する。
- [0067] (ステップS 3 0 9) 変数情報取得手段 1 4 2 は、変換表にレコードを追記する。レコードは、変数の 1 以上の入力値、または変数の 1 以上の入力値と変数の 1 以上の出力値を有する。
- [0068] (ステップS 3 1 0) 変数情報取得手段 1 4 2 は、カウンタ i を 1、インクリメントする。ステップS 3 0 3に戻る。
- [0069] (ステップS 3 1 1) 変数情報取得手段 1 4 2 は、型を用いて、変数の 1 以上の入力値、または変数の 1 以上の入力値と変数の 1 以上の出力値を取得する。ステップS 3 0 9に行く。なお、変数情報取得手段 1 4 2 は、例えば、型と 1 以上の値との対応表を保持しており、かかる対応表を用いて、型から 1 以上の入力値、または 1 以上の入力値と 1 以上の出力値とを取得する。
- [0070] なお、図 3 のフローチャートにおいて、例えば、デフォルトの変数の情報（例えば、後述する" null#true" など）からレコードを構成し、当該レコードを変数表に書き込んでも良い。かかる場合、変数情報取得手段 1 4 2 は、デフォルトの変数の情報を予め保持している。
- [0071] 次に、ステップS 2 0 4 の命題論理式取得処理について、図 4 のフローチャートを用いて説明する。

- [0072] (ステップS 4 0 1) テスト情報取得手段 1 4 3 は、カウンタ  $i$  に 1 を代入する。
- [0073] (ステップS 4 0 2) テスト情報取得手段 1 4 3 は、仕様記述情報から  $i$  番目の処理単位を読み出す。
- [0074] (ステップS 4 0 3) テスト情報取得手段 1 4 3 は、ステップS 4 0 2 で  $i$  番目の処理単位を読み出せたか否かを判断する。 $i$  番目の処理単位を読み出せればステップS 4 0 4 に行き、読み出せなければ上位処理にリターンする。
- [0075] (ステップS 4 0 4) テスト情報取得手段 1 4 3 は、 $i$  番目の処理単位が制約であるか否かを判断する。制約であればステップS 4 0 5 に行き、制約でなければステップS 4 0 6 に行く。なお、制約とは、仕様記述における、対象が満たさなければならない条件である。テスト情報取得手段 1 4 3 は、例えば、予約語（例えば、"Require"）に続く行であり、次の制約以外を示す予約語が出現するまでの行を、制約情報として判断する。ただし、制約情報の判断手法は問わない。
- [0076] (ステップS 4 0 5) テスト情報取得手段 1 4 3 は、 $i$  番目の処理単位に対応する制約情報を、命題論理式に変換する。なお、制約情報を命題論理式に変換する技術は公知技術である。
- [0077] (ステップS 4 0 6) テスト情報取得手段 1 4 3 は、カウンタ  $i$  を 1、インクリメントする。ステップS 4 0 2 に戻る。
- [0078] 次に、ステップS 2 0 5 のテスト情報取得処理について、図 5 のフローチャートを用いて説明する。
- [0079] (ステップS 5 0 1) テスト情報取得手段 1 4 3 は、カウンタ  $i$  に 1 を代入する。
- [0080] (ステップS 5 0 2) テスト情報取得手段 1 4 3 は、 $i$  番目の入力値のセットを、変数表から取得する。テスト情報取得手段 1 4 3 は、1 以上の各変数が入力値として採り得る値の中から、各変数の値を取得し、変数名と値の組を有する 1 以上の情報を取得する。この変数名と値の組を有する 1 以上の

情報が、入力値のセットである。

- [0081] (ステップS 5 0 3) テスト情報取得手段 1 4 3 は、ステップS 5 0 2 で i 番目の入力値のセットを取得できたか否かを判断する。i 番目の入力値のセットを取得できればステップS 5 0 4 に行き、取得できなければ上位処理にリターンする。
- [0082] (ステップS 5 0 4) テスト情報取得手段 1 4 3 は、カウンタ j に 1 を代入する。
- [0083] (ステップS 5 0 5) テスト情報取得手段 1 4 3 は、j 番目の出力値のセットを、変数表から取得する。テスト情報取得手段 1 4 3 は、1 以上の各変数が出力値として採り得る値の中から、各変数の値を取得し、変数名と値の組を有する 1 以上の情報を取得する。この変数名と値の組を有する 1 以上の情報が、出力値のセットである。
- [0084] (ステップS 5 0 6) テスト情報取得手段 1 4 3 は、ステップS 5 0 5 で j 番目の出力値のセットを取得できたか否かを判断する。j 番目の出力値のセットを取得できればステップS 5 0 7 に行き、取得できなければステップS 5 1 6 に行く。
- [0085] (ステップS 5 0 7) テスト情報取得手段 1 4 3 は、カウンタ k に 1 を代入する。
- [0086] (ステップS 5 0 8) テスト情報取得手段 1 4 3 は、k 番目の命題論理式を取得する。ここでの命題論理式は、ステップS 2 0 4 の命題論理式取得処理により取得された 1 以上の命題論理式の中の k 番目の命題論理式である。
- [0087] (ステップS 5 0 9) テスト情報取得手段 1 4 3 は、ステップS 5 0 8 で k 番目の命題論理式を取得できたか否かを判断する。k 番目の命題論理式を取得できればステップS 5 1 0 に行き、k 番目の命題論理式を取得できなければステップS 5 1 3 に行く。
- [0088] (ステップS 5 1 0) テスト情報取得手段 1 4 3 は、i 番目の入力値と j 番目の出力値とを、k 番目の命題論理式に適用する。
- [0089] (ステップS 5 1 1) テスト情報取得手段 1 4 3 は、i 番目の入力値と j

番目の出力値とが、k番目の命題論理式を満たすか否かを判断する。k番目の命題論理式を満たせばステップS512に行き、満たさなければステップS515に行く。

[0090] (ステップS512) テスト情報取得手段143は、カウンタkを1、インクリメントする。ステップS508に戻る。

[0091] (ステップS513) テスト情報取得手段143は、i番目の入力値とj番目の出力値とを有するテスト情報を構成する。

[0092] (ステップS514) テスト情報取得手段143は、ステップS513で構成したテスト情報を図示しないバッファに追記する。このバッファは、通常、一時的な記憶媒体である。

[0093] (ステップS515) テスト情報取得手段143は、カウンタjを1、インクリメントする。ステップS505に戻る。

[0094] (ステップS516) テスト情報取得手段143は、カウンタiを1、インクリメントする。ステップS502に戻る。

[0095] 以下、本実施の形態におけるテスト仕様生成装置1の具体的な動作について説明する。

[0096] 今、仕様記述情報格納部12は、図6に示す仕様記述情報を格納している。図6において、(1)(2)は、変数の型の宣言である。(1)のデータの型“onoff\_type”の変数は、値“ON”または“OFF”を採り得る。データの型“onoff\_type”が値“ON”または“OFF”を採り得るという情報は、テスト仕様生成装置1が保持している。つまり、例えば、テスト仕様生成装置1は「onoff\_type : on | off」を保持している。また、(2)は、データの型“m\_type”の初期値は“m”であることを示す。

[0097] また、図6において、(3)は、変数の宣言の例である。(3)において、テスト元情報の予約語“testedwith”により、テスト元情報{off}が存在することが分かる。テスト元情報{off}は、型“onoff\_type”の変数“op”のテストの際に利用する値として、“off”のみであることを示す。つまり、このテスト元情報の記載により、仕様記述情報の作成者の、テストにおける

意図が反映できる。なお、“op:onoff\_type{off}”の“off”は、変数“op”の初期値が“off”であることを示す。

[0098] また、図6において、(4)も変数の宣言の例であるが、テスト元情報は有さない。そして、変数“lamp”の型は“onoff\_type”であり、初期値は“off”である。かかる場合、変数“lamp”の値の中で、テストで利用する値は、変数“lamp”が採り得る値のすべて（ここでは、“on”と“off”）である。

[0099] また、図6の(5)は、型“m\_type”の変数“ch”の宣言である。

[0100] また、図6の(6)(7)は、制約情報である。予約語“Function”、“Require”が、以降に続く文字列が制約情報であることを示す。また、(6)は、状態遷移情報である。(6)において、“op==on && lamp==off”は遷移前状態、“ch?m”はイベント、“lamp==on”は遷移後状態を示す。つまり、(6)は、“op==on && lamp==off”の場合に、イベント“ch?m”が発生すれば、“lamp==on”に状態が変化していなければならないという制約を示す。また、(7)は不変条件情報である。(7)において、“op==off -> lamp==off”は、“op==off”であれば、“lamp==off”であることを示す。つまり、“A->B”は、「AならばB」であることを示す。

[0101] かかる状況において、ユーザは、テスト仕様生成装置1に、テスト情報生成指示を入力した、とする。

[0102] すると、受付部13は、テスト情報生成指示を受け付ける。そして、テスト情報取得部14は、仕様記述情報格納部12から、図6の仕様記述情報を読み出す。

[0103] 次に、テスト情報取得部14の変数情報取得手段142は、図6の仕様記述情報を解釈し、変数表を取得する。つまり、変数情報取得手段142は、まず、デフォルトの変数の情報“null#true”、“SubsystemCountUp#true”を、図示しない記録媒体（通常、変数情報取得手段142が保持する記録媒体）から取得する。そして、変数情報取得手段142は、取得した各変数の情報に変数の識別符号（以下、変数識別子という）を付してレコードを構成

し、当該レコードを空の変数表に書き込む。なお、ここでは、変数識別子は、「1」から順次、インクリメントされる整数である、とする。つまり、変数情報取得手段142は、まず、レコード”1 null#true”、”2 SubsystemCountUp#true”を変数表に書き込む。

[0104] 次に、変数情報取得手段142は、図6の”Extern”、”Data onoff\_type ; // from main”、”Class B”、”Data m\_type {m};”を順次、読み出し、変数の宣言ではない、と判断する。なお、ここでの処理単位は、例えば、行である、とする。

[0105] 次に、変数情報取得手段142は、図6の”Var op : onoff\_type {off} testedwith {off};”を読み出す。そして、変数情報取得手段142は、予約語”Var”から、この処理単位が変数の宣言である、と判断する。なお、以降に続く処理単位に対して、変数ではない予約語が出現するまで、変数情報取得手段142は、変数の宣言であると判断する、とする。

[0106] 次に、変数情報取得手段142は、処理単位”Var op : onoff\_type {off} testedwith {off};”から変数の型”onoff\_type”を取得する。次に、テスト元情報取得手段141は、この処理単位からテスト元情報”testedwith{off}”を取得する。

[0107] 次に、変数情報取得手段142は、テスト元情報”testedwith{off}”を用いて、変数”op”の1以上の入力値、または変数の1以上の入力値と変数の1以上の出力値（次の状態の値）を取得する。つまり、変数情報取得手段142は、変数”op”の入力値”off”、および変数”op”の出力値”off”を取得する。つまり、テスト元情報”testedwith{off}”は、変数”op”のテストの値として利用されるのは”off”だけであり、”on”は利用されないことを示す。そして、変数情報取得手段142は、変数”op”の入力値”off”に変数識別子を付して、レコード”3 op#off”を構成する。なお、”3 op#off”のうちの”op#off”の前に”@”が存在しない場合、それは入力値であることを示す。入力値は、状態の遷移前の値である。また、変数情報取得手段142は、変数”op”の出力値”off”に変数識別子を付して、レコード”

4 @op#off”を構成する。なお、“4 @op#off”のうちの“@”は出力値であることを示す。

[0108] 次に、変数情報取得手段142は、図6の“lamp : onoff\_type {off};”を読み出す。そして、変数情報取得手段142は、この処理単位が変数の宣言である、と判断する。

[0109] 次に、変数情報取得手段142は、処理単位“lamp : onoff\_type {off};”から変数の型“onoff\_type”を取得する。次に、テスト元情報取得手段141は、この処理単位からテスト元情報を取得できない。

[0110] 次に、変数情報取得手段142は、型“onoff\_type”を用いて、変数“lamp”の1以上の入力値{on, off}と変数“lamp”の1以上の出力値{on, off}を取得する。そして、変数情報取得手段142は、変数“lamp”の入力値“on”に変数識別子を付して、レコード“5 lamp#on”を構成する。また、変数情報取得手段142は、変数“lamp”の入力値“off”に変数識別子を付して、レコード“6 lamp#off”を構成する。また、変数情報取得手段142は、変数“lamp”の出力値“on”に変数識別子を付して、レコード“7 @lamp#on”を構成する。さらに、変数情報取得手段142は、変数“lamp”の出力値“off”に変数識別子を付して、レコード“8 @lamp#off”を構成する。そして、変数情報取得手段142は、これらのレコードを変換表に追記する。

[0111] 次に、変数情報取得手段142は、図6の“Channel ch : m\_type;”を読み出す。そして、変数情報取得手段142は、予約語“Channel”から、この処理単位が変数の宣言である、と判断する。

[0112] 次に、変数情報取得手段142は、処理単位“Channel ch : m\_type;”から変数の型“m\_type”を取得する。次に、テスト元情報取得手段141は、この処理単位からテスト元情報を取得できない。

[0113] 次に、変数情報取得手段142は、型“m\_type”を用いて、変数“ch”の入力値{m}を取得する。そして、変数情報取得手段142は、変数“ch”の入力値“m”に変数識別子を付して、レコード“9 ch?m”を構成する。そして

、変数情報取得手段 1 4 2 は、変換表にこのレコードを追記する。なお、予約語” Channel” はイベントを示し、入力値のみ存在し、出力値が存在しないことは、予め変数情報取得手段 1 4 2 は知っている。

[0114] 以上の処理により、図 7 に示す変数表が得られた。変数表において、” c --- CNFNum to VarAssign table --- “、” c --- end --- “、および行頭の” c” は、ここでは重要な意味を持たない、とする。

[0115] 次に、テスト情報取得手段 1 4 3 は、図 6 の仕様記述情報を解釈し、以下のように命題論理式を取得する。

[0116] つまり、テスト情報取得手段 1 4 3 は、例えば、処理単位” Var op : onoff\_type {off} testedwith{off};” を読み出し、当該処理単位から状態遷移前の命題論理式「op#off」と、状態遷移後の命題論理式「@op#off」とを得る。テスト情報取得手段 1 4 3 は、処理単位” Var op : onoff\_type {off} testedwith {off};” を解釈し、状態遷移前の変数” op” の値は常に” off” であり、状態遷移後の変数” op” の値は常に” off” であると決定する。なお、「op#off」は、状態遷移前の変数” op” の値が” off” であることを示す。また、「@op#off」は、状態遷移後の変数” op” の値が” off” であることを示す。

[0117] そして、ここでは、テスト情報取得手段 1 4 3 は、命題論理式「op#off」に対して、図 7 の変数表を用いて、「op#off」に対応する変数識別子「3」を取得する。そして、「op#off」を「3 0」に置き換える。なお、「3 0」の「0」は、命題論理式の終端を示す。また、命題論理式「op#off」をコメントとして付加する。図 8 が、テスト情報取得手段 1 4 3 が取得した命題論理式の集合の例である。図 8 において、命題論理式はコメントアウトされ（行頭にcが付加され）、それと同意義の変数識別子の集合に置き換わっている。図 8 の（1）が「op#off」に対応し、（2）がそれと同意義の変数識別子の集合に対応する。同様に、図 8 の（3）が「@op#off」に対応し、（4）がそれと同意義の変数識別子の集合に対応する。なお、テスト情報取得手段 1 4 3 は、コメントアウトされた命題論理式を付加しなくとも良い。

[0118] また、テスト情報取得手段 143 は、例えば、処理単位” op==on&&lamp==off:ch?m: lamp==on;” を読み出し、当該処理単位から命題論理式「 $op\#on \wedge lamp\#off \wedge ch?m \rightarrow @lamp\#on$ 」を得る。つまり、テスト情報取得手段 143 は、” op==on” を「 $op\#on$ 」、” lamp==off” を「 $lamp\#off$ 」、” ch?m” を「 $ch?m$ 」にし、” &&” を「 $\wedge$ 」に変換し、遷移前状態イベントとを「 $\wedge$ 」で連結する。そして、遷移後状態” lamp==on” を「 $@lamp\#on$ 」に変換し、「 $\rightarrow$ 」で連結する。以上の処理により、テスト情報取得手段 143 は、仕様記述情報から命題論理式を得る。

[0119] 次に、テスト情報取得手段 143 は、処理単位” op==on&&lamp==off:ch?m: lamp==on;” をコメントアウトし、上述のように、当該命題論理式「 $op\#on \wedge lamp\#off \wedge ch?m \rightarrow @lamp\#on$ 」を変数識別子の集合に変換する処理を行う。ここで、「 $op\#on$ 」に対応する変数識別子の変数表に存在しないので、テスト情報取得手段 143 は、命題論理式「 $op\#on \wedge lamp\#off \wedge ch?m \rightarrow @lamp\#on$ 」を変数識別子の集合に変換できない、と判断する。よって、テスト情報取得手段 143 は、図 8 の (5) の情報を得る。

[0120] 次に、テスト情報取得手段 143 は、処理単位” op==off  $\rightarrow$  lamp==off;” を読み出し、当該処理単位から状態遷移前の命題論理式「 $\neg op\#off \wedge lamp\#off$ 」と、状態遷移後の命題論理式「 $\neg @op\#off \wedge @lamp\#off$ 」とを得る。なお、「 $\neg$ 」は否定を示す。

[0121] そして、テスト情報取得手段 143 は、処理単位” op==off  $\rightarrow$  lamp==off;” の遷移前の状態をコメントアウトし、図 8 の (6) を得る。

[0122] そして、テスト情報取得手段 143 は、状態遷移前の命題論理式「 $\neg op\#off \wedge lamp\#off$ 」を変数識別子の集合に変換する処理を行い、「-3 6 0」を得る。そして、テスト情報取得手段 143 は、図 8 の (7) を得る。ここで「-3」の「-」は否定を示す。つまり、「 $\neg op\#off$ 」から、変数表の「3 (op#off)」をマイナスした値である「-3」を得る。また、テスト情報取得手段 143 は、「 $lamp\#off$ 」を変数表に適用し、「6」を得る。

[0123] 次に、テスト情報取得手段 143 は、処理単位” op==off  $\rightarrow$  lamp==off;”

の遷移後の状態をコメントアウトし、図 8 の (8) を得る。そして、テスト情報取得手段 143 は、状態遷移後の命題論理式「 $\neg \text{op}\#\text{off} \wedge \text{lamp}\#\text{off}$ 」を変数識別子の集合に変換する処理を行い、「-4 8 0」を得る。そして、テスト情報取得手段 143 は、図 8 の (9) を得る。つまり、「 $\neg \text{op}\#\text{off}$ 」から、変数表の「4 (@op#off)」をマイナスした値である「-4」を得る。また、テスト情報取得手段 143 は、「@lamp#off」を変数表に適用し、「8」を得る。

[0124] 以上のような処理を、仕様記述情報のすべての処理単位に対して行い、テスト情報取得手段 143 は、図 8 の乗法標準形 (CNF) のファイルを得る。

[0125] 次に、テスト情報取得手段 143 は、状態遷移前の変数”op”の採り得る値”off”と”on”、状態遷移後の変数”@op”の採り得る値”off”と”on”、状態遷移前の変数”lamp”の採り得る値”off”と”on”、状態遷移後の変数”@lamp”の採り得る値”off”と”on”、チャンネルの変数の採り得る値”m”、デフォルトの”null”の採り得る値”true”と”false”、デフォルトのカウンタ”SubsystemCountUp”の採り得る値”true”と”false”、とすべての組み合わせ（入力値と出力値のセット）が、図 8 のすべての命題論理式（図 8 は、正確には命題論理式と同等の変数識別子の集合である）を満足するか否かを判断する。そして、テスト情報取得手段 143 は、すべての命題論理式を満足する入力値と出力値のセットを取得する。この入力値と出力値のセットが、テスト情報である。ただし、ここでは、テスト情報取得手段 143 は、入力値と出力値のセットを、変数表を参照して、数値列に変換する。ここで、テスト情報取得手段 143 が取得した数値列が、図 9 の (1) (2) (3) である。なお、図 9 (1) の「-1 2 3 4 -5 6 -7 8 -9」の「-1」は、変数表の「1 null#true」の否定を示す。また、図 9 (1) の「2」は、変数表の「2 SubsystemCountUp#true」を示す。また、図 9 (1) の「3」は、変数表の「3 op#off」を示す。

[0126] 次に、テスト情報取得手段 143 は、図 9 (1) から (3) の数値列を、

(図7)を用いて、入力値と出力値のセットに変換し、図10のテスト情報を得る。図10がテスト情報の一例であり、ここではテストスイートである。なお、図10において、actionの属性は存在しなくても良い。actionの属性は、状態遷移後に明示的に発生させるイベントが代入され得るが、ここでは、すべて”null”である。

[0127] 以下に、テスト情報取得部14の動作の概念について、図11を用いて、さらに説明する。図11において、テスト情報取得部14は、T1、T2、T3の3つの処理を行う。

[0128] T1において、仕様記述情報（ここでは、特に、SENSという仕様記述言語で記述したファイルであるSENS記述ファイル）が入力され、CNFファイルが出力される。詳細には、SENS記述に現れる制約情報が乗法標準形（CNF）の命題論理式に変換され、その命題論理式を表す数列の組が出力される。処理の手順は以下の通りである。なお、以下の処理は、すべてテスト情報取得部14が行う処理である。

[0129] 1. テスト情報取得部14は、SENS記述中に現れる変数について、その変数が現時刻でどの値をとるかを原子命題で表現し、それら原子命題に対してユニークな整数値を割り当てる。この整数値は原子命題の通し番号であり、その原子命題の真偽値が真であることはその整数値の正の値で、偽であることは負の値で表現される。同様に次の時刻でその変数がどの値をとるかを原子命題で表し、その原子命題に対してもユニークな整数値を割り当てる。

[0130] 例えば、on, offの値を持つSENSの変数lampについて、テスト情報取得部14は、現時刻でlampがonであることを表す命題をlamp#on, offであることを表す命題をlamp#offとし、それぞれ整数値5, 6を割り当てる。同様に次の時刻でlampがonであることを表す命題を@lamp#on, offであることを表す命題を@lamp#offとし、それぞれ整数値7, 8を割り当てる（図8参照）。

[0131] 2. テスト情報取得部14は、SENS記述中に現れる制約情報を命題論理式に変換する。「事前条件：イベント：事後条件, アクション」の形で表現される制約情報は「事前条件∧イベント → 事後条件∧アクション」といった形

の命題論理式に変換される。このとき事後条件に現れる変数については、次の時刻の表現として扱う。例えば、SENS記述に現れる「 $op==on \ \&\& \ lamp==off$  :  $ch?m$  :  $lamp==on$  ;」という記述は、命題論理式「 $op\#on \wedge lamp\#off \wedge ch?m \rightarrow @lamp\#on$ 」に変換される。

[0132] 3. テスト情報取得部 14 は、制約情報を表現した命題論理式を乗法標準形に変換する。例えば、命題論理式「 $op\#on \wedge lamp\#off \wedge ch?m \rightarrow @lamp\#on$ 」を乗法標準形に変換すると「 $\neg op\#on \vee \neg lamp\#off \vee \neg ch?m \vee @lamp\#on$ 」となる。

[0133] 4. 乗法標準形の節は、1. で原子命題に割り当てた整数値を使って数列で表現される。このとき否定のリテラルは負の数で表される。テスト情報取得部 14 は、一つの節に対する数列を一行として、ファイルに出力する。例えば、節「 $\neg op\#on \vee \neg lamp\#off \vee \neg ch?m \vee @lamp\#on$ 」は、通常、数列に変換されるが、ここでは、テスト元情報により” op” は” off” のみしかテスト情報の対象とならないので、数列に変換できず、数列は出力されない。

[0134] 5. テスト情報取得部 14 は、原子命題と整数値の対応表をファイルの末尾にコメントとして書き出す。

[0135] 次に、T2において、テスト情報取得部 14 は、CNFファイルを受け付け、DNFファイルを出力する。つまり、CNFファイルが表現している命題論理式を満足する真偽値割り当ては、SAT-solver（公知技術によるソフトウェア）を用いてすべて探索される。処理の流れは以下の図12の通りである。

[0136] SAT-solverは値割り当てを加法標準形（DNF）で出力する。その値割り当てを、（a）数列表現でファイルに書き出し、（b）同時にその論理否定をSAT-solverにフィードバックし、次の値割り当てを得る。これを値割り当てが出力されなくなるまで繰り返す。T2の出力ファイル中に現れる各数列は、CNFファイルが表現する論理式を満足する値割り当ての一つを表現している。

[0137] 次に、T3の処理を行う。T3において、SENS記述から作成したCNFファイルとDNFファイルを入力し、CSVファイルを出力する。

[0138] DNFファイルの各数列をCNFファイル末尾の対応表を基に論理式と解釈し、S

ENSの変数とその値の割り当て表を表現するCSV形式に変換する。CSVの一行がテスト実行時の入力と、それに対応する期待される出力を表す。すなわちCSVの一行が一组のテストスイートを表す。例えば、DNFファイルの一行「-1 -2 3 4 -5 6 -7 8 9」を、命題論理式「 $\text{op}\#\text{off}\wedge\text{lamp}\#\text{off}\wedge\text{ch?m}\wedge\text{@op}\#\text{off}\wedge\text{@lamp}\#\text{off}$ 」と解釈して、CSV形式「off, off, ch?m, off, off, null」に変換する（図10の第二レコードを参照）。

[0139] 以上、本実施の形態によれば、対象の設計時にテスト設計のための情報を記載すれば、適切なテスト仕様に関する情報が自動生成できる。

[0140] また、本実施の形態によれば、仕様記述情報の中にテスト元情報が記述できることにより、設計者の意図を反映したテスト情報の生成ができる。

[0141] なお、本実施の形態における処理は、ソフトウェアで実現しても良い。そして、このソフトウェアをソフトウェアダウンロード等により配布しても良い。また、このソフトウェアをCD-ROMなどの記録媒体に記録して流布しても良い。なお、このことは、本明細書における他の実施の形態においても該当する。なお、本実施の形態におけるテスト仕様生成装置を実現するソフトウェアは、以下のようなプログラムである。つまり、このプログラムは、記憶媒体に、テストで利用する情報であるテスト情報と、対象の仕様について記述する言語である仕様記述言語で記述された前記対象の仕様に関する情報であり、1以上の変数の宣言を含む情報であり、テスト情報を生成するためのテスト元情報を含む情報である仕様記述情報を格納しており、コンピュータを、前記変数の宣言および前記テスト元情報を用いて、1以上の各変数の入力値を少なくとも有する1以上のテスト情報を取得するテスト情報取得部と、前記テスト情報取得部が取得した1以上のテスト情報を前記テスト情報格納部に蓄積するテスト情報蓄積部として機能させるためのプログラム、である。

[0142] また、上記プログラムにおいて、前記変数は、型を有し、前記仕様記述情報は、前記1以上の変数のうち、1以上の各変数に対応付けて、各変数が採り得る値の中で、テストで使用する値を特定する情報であるテスト元情報を

有し、前記テスト情報取得部は、前記仕様記述情報から、前記テスト元情報を前記変数と対応付けて取得するテスト元情報取得手段と、前記仕様記述情報から、テスト元情報を有さない変数と当該変数の型を取得する変数情報取得手段と、前記テスト元情報取得手段が取得したテスト元情報から、各テスト元情報に対応する変数の1以上の入力値を変数ごとに取得し、かつ、前記変数情報取得手段が取得した変数と当該変数の型から入力値を変数ごとに取得するテスト情報取得手段を具備するものとして、コンピュータを機能させるためのプログラムであることは好適である。

[0143] また、上記プログラムにおいて、前記テスト情報取得手段は、前記テスト元情報取得手段が取得したテスト元情報から、各テスト元情報に対応する変数の1以上の入力値および出力値を変数ごとに取得し、かつ、前記変数情報取得手段が取得した変数と当該変数の型から入力値および出力値を変数ごとに取得するものとして、コンピュータを機能させるためのプログラムであることは好適である。

[0144] また、上記プログラムにおいて、前記仕様記述情報は、1以上の変数の宣言、および制約に関する記述である1以上の制約情報を有し、前記テスト情報取得手段は、前記テスト元情報取得手段が取得したテスト元情報から、各テスト元情報に対応する変数の1以上の入力値および出力値を変数ごとに取得し、かつ、前記変数情報取得手段が取得した変数と当該変数の型から入力値および出力値を変数ごとに取得し、かつ、前記すべての制約情報を満たすように、1以上の入力値および1以上の出力値の組であるテストスイートを1以上取得するものとして、コンピュータを機能させるためのプログラムであることは好適である。

[0145] また、上記プログラムにおいて、前記制約情報は、1以上の変数の値の集合を有する状態である遷移前状態と当該遷移前状態に対して発生するイベントと前記状態の場合に前記イベントが発生した場合に移行する状態である遷移後状態とを有する状態遷移情報、または1以上の変数の値の集合を有する第一状態と当該第一状態の場合に必ず採用される1以上の変数の値の集合を

有する第二状態とを有する不変条件情報であるものとして、コンピュータを機能させるためのプログラムであることは好適である。

[0146] (実施の形態 2)

本実施の形態において、不適切なテストスイートの生成を防止できるテスト仕様生成装置 2 について説明する。本実施の形態において、第一に、状態遷移情報の前提部分を制約として、状態遷移の前提を満たさないテストスイートの生成を防止するテスト仕様生成装置 2 について説明する。第二に、イベントに関する制約を設け、指定したイベントを含むテストスイートの生成を防止するテスト仕様生成装置 2 について説明する。第三に、遷移前状態情報を削除し、残りの部分の論理否定を制約情報とするテスト仕様生成装置 2 について説明する。第四に、遷移後状態情報を削除し、残りの部分の論理否定を制約情報とするテスト仕様生成装置 2 について説明する。

[0147] 図 13 は、本実施の形態におけるテスト仕様生成装置 2 のブロック図である。テスト仕様生成装置 2 は、テスト情報格納部 11、仕様記述情報格納部 12、受付部 13、テスト情報取得部 24、テスト情報蓄積部 15、テスト情報出力部 16 を備える。

[0148] テスト情報取得部 24 は、イベント制約情報格納手段 240、テスト元情報取得手段 141、変数情報取得手段 142、テスト情報取得手段 243 を備える。

[0149] 仕様記述情報格納部 12 が格納している仕様記述情報は、ここでは、1 以上の変数の宣言、および 1 以上の制約情報を有する。また、制約情報は、ここでは、例えば、遷移前状態とイベントと前記状態の場合に遷移後状態とを有する状態遷移情報である。状態遷移情報は、アクションを有しても良い。また、制約情報は、例えば、不変条件情報である。

[0150] テスト情報取得部 24 は、変数の宣言を用いて、1 以上の各変数の入力値を少なくとも有する 1 以上のテスト情報を取得する。また、テスト情報取得部 24 は、変数の宣言とテスト元情報を用いて、1 以上の各変数の入力値を少なくとも有する 1 以上のテスト情報を取得することは好適である。また、

ここでのテスト情報は、通常、テストスイートである。

- [0151] イベント制約情報格納手段 240 は、1 以上のイベント制約情報を格納し得る。イベント制約情報は、イベントに関する制約を示す情報である。イベント制約情報は、例えば、「 $\neg \text{null}$ 」、「 $\neg t?\text{clear}$ 」などである。「 $\neg \text{null}$ 」は、イベントは  $\text{null}$  ではないことを示す。また、「 $\neg t?\text{clear}$ 」は、イベントは「 $t?\text{clear}$ 」ではないことを示す。
- [0152] テスト情報取得部 24 は、通常、MPU やメモリ等から実現され得る。テスト情報取得部 24 の処理手順は、通常、ソフトウェアで実現され、当該ソフトウェアは ROM 等の記録媒体に記録されている。但し、ハードウェア（専用回路）で実現しても良い。
- [0153] テスト情報取得手段 243 は、仕様記述情報から、変数情報取得手段 142 が取得した変数と当該変数の型から入力値および出力値を変数ごとに取得し、かつ、すべての制約情報を満たすように、1 以上の入力値および 1 以上の出力値の組であるテストスイートを 1 以上取得する。また、テスト情報取得手段 243 は、仕様記述情報から、各テスト元情報に対応する変数の 1 以上の入力値および出力値を変数ごとに取得し、かつ、すべての制約情報を満たすように、1 以上の入力値および 1 以上の出力値の組であるテストスイートを 1 以上取得しても良い。
- [0154] テスト情報取得手段 243 は、1 以上の前提情報をも制約情報として用いて、すべての制約情報を満たすように、1 以上の入力値および 1 以上の出力値の組であるテストスイートを 1 以上取得することは好適である。ここで、前提情報とは、1 以上の制約情報が有する遷移前状態とイベントとの組である。前提情報は、遷移前状態のみでも良い。また、通常、1 以上の前提情報の論理和が制約情報となる。仕様記述情報が有する元の制約情報が「 $(\text{op}=\text{on} \wedge \text{null}) \rightarrow t!\text{start}; (\text{op}=\text{on} \wedge t?\text{ring}) \rightarrow \text{arg1.ch!m} \wedge \text{arg2.ch!m};$ 」である場合、テスト情報取得手段 243 は、当該制約情報の前提情報「 $(\text{op}=\text{on} \wedge \text{null}) \vee (\text{op}=\text{on} \wedge t?\text{ring})$ 」を取得し、制約情報に追加する。ここで、「 $(\text{op}=\text{on} \wedge \text{null}) \rightarrow t!\text{start}$ 」は、変数「 $\text{op}$ 」が「 $\text{on}$ 」かつ「 $\text{null}$ 」であればチャネ

ル「t」にメッセージ「start」を送信する、ことを示す。また、「(op==on/\nt?ring)-> arg1.ch!m\arg2.ch!m;」は変数「op」が「on」かつチャンネル「t」が「ring」を受信すれば、チャンネル「arg1.ch」にメッセージ「m」かつチャンネル「arg2.ch」にメッセージ「m」を送信することを示す。

[0155] また、テスト情報取得手段243は、1以上のイベント制約情報をも制約情報として用いて、すべての制約情報を満たすように、1以上の入力値および1以上の出力値の組であるテストスイートを1以上取得することは好適である。ここで、イベント制約情報は、イベント制約情報格納手段240のイベント制約情報である。

[0156] また、テスト情報取得手段243は、既に取得したテストスイート（解と言っても良い。）に対応する制約情報から、1以上の第二制約情報を取得し、当該1以上の第二制約情報をも制約情報として用いて、すべての制約情報を満たすように、1以上の入力値および1以上の出力値の組であるテストスイートを1以上取得することは好適である。ここで、第二制約情報は、既に取得したテストスイートの情報が有する遷移前状態情報を削除したうえで、残りの部分の論理否定をとることにより構成される情報である。また、第二制約情報は、制約情報が有する遷移前状態およびイベントを削除したうえで、残りの部分（遷移後状態）の論理否定をとることにより構成される情報であっても良い。例えば、解「op#off, null#true, @op#off」が得られた場合、テスト情報取得手段243は、第二制約情報「@op#offでないもの」（「@op#off」）を、制約情報として追加する。なお、遷移前状態を削除することは、制約情報から、遷移前状態以外の部分を取得することと同意義である。

[0157] また、第二制約情報は、制約情報が有する遷移後状態情報を削除して、かつ、当該制約情報が有する情報であり、前記遷移後状態以外の部分の論理否定により構成される情報であっても良い。また、第二制約情報は、制約情報が有するアクションおよび遷移後状態を削除して、かつ、当該制約情報が有する情報であり、前記削除した部分以外の部分（遷移前状態）の論理否定により構成される情報であっても良い。例えば、解「op#off, null#true, @op#of

f」が得られた場合、テスト情報取得手段243は、第二制約情報「op#offでないイベントがnullでない」( $\neg(\text{op\#off} \parallel \text{null\#true})$ )を、制約情報として追加する。なお、遷移後状態を削除することは、制約情報から、遷移後状態以外の部分を取得することと同意義である。

[0158] テスト情報取得手段243は、通常、MPUやメモリ等から実現され得る。テスト情報取得手段243の処理手順は、通常、ソフトウェアで実現され、当該ソフトウェアはROM等の記録媒体に記録されている。但し、ハードウェア（専用回路）で実現しても良い。

[0159] 次に、本実施の形態におけるテスト仕様生成装置2の具体的な動作について説明する。以下、4つの具体例について説明する。

[0160] （具体例1）

具体例1は、1以上の前提情報をも制約情報として用いて、無用なテストスイートの抑制を図る場合である。

[0161] 今、仕様記述情報格納部12は、図14に示す仕様記述情報を格納している。図14の仕様記述情報は、図6の仕様記述情報と比較して、(3)の「testedwith{off}」を有しない。図14において、図6と同様、制約情報は(6)(7)である。

[0162] かかる状況において、ユーザは、テスト仕様生成装置1に、テスト情報生成指示を入力した、とする。

[0163] すると、受付部13は、テスト情報生成指示を受け付ける。そして、テスト情報取得部14は、仕様記述情報格納部12から、図14の仕様記述情報を読み出す。

[0164] 次に、テスト情報取得部14の変数情報取得手段142は、図14の仕様記述情報を解釈し、変数表を取得する。つまり、変数情報取得手段142は、まず、デフォルトの変数の情報”null#true”、”SubsystemCountUp#true”を、図示しない記録媒体から取得する。そして、変数情報取得手段142は、取得した各変数の情報に変数識別子を付してレコードを構成し、当該レコードを空の変数表に書き込む。

[0165] 次に、変数情報取得手段142は、図14の”Extern”、”Data onoff\_type; // from main”、”Class B”、”Data m\_type {m};”を順次、読み出し、変数の宣言ではない、と判断する。

[0166] 次に、変数情報取得手段142は、図6の”Var op : onoff\_type {off};”を読み出す。そして、変数情報取得手段142は、予約語”Var”から、この処理単位が変数の宣言である、と判断する。なお、以降に続く処理単位に対して、変数ではない予約語が出現するまで、変数情報取得手段142は、変数の宣言であると判断する、とする。

[0167] 次に、変数情報取得手段142は、処理単位”Var op : onoff\_type {off};”から変数の型”onoff\_type”を取得する。次に、テスト元情報取得手段141は、この処理単位からテスト元情報を取得できない。

[0168] 次に、変数情報取得手段142は、変数の型”onoff\_type”から、変数”op”の入力値”on”と”off”、および変数”op”の出力値”on”と”off”を取得する。そして、変数情報取得手段142は、変数”op”の入力値”on”に変数識別子を付して、レコード”3 op#on”を構成する。また、変数情報取得手段142は、変数”op”の入力値”off”に変数識別子を付して、レコード”4 op#off”を構成する。また、変数情報取得手段142は、変数”op”の出力値”on”に変数識別子を付して、レコード”5 @op#on”を構成する。また、変数情報取得手段142は、変数”op”の出力値”off”に変数識別子を付して、レコード”6 @op#off”を構成する。

[0169] 次に、変数情報取得手段142は、図6の”lamp : onoff\_type {off};”を読み出す。そして、変数情報取得手段142は、この処理単位が変数の宣言である、と判断する。

[0170] 次に、変数情報取得手段142は、処理単位”lamp : onoff\_type {off};”から変数の型”onoff\_type”を取得する。次に、テスト元情報取得手段141は、この処理単位からテスト元情報を取得できない。

[0171] 次に、変数情報取得手段142は、型”onoff\_type”を用いて、変数”lamp”の1以上の入力値{on, off}と変数”lamp”の1以上の出力値{on, off}を

取得する。そして、変数情報取得手段 142 は、変数” lamp” の入力値” on” に変数識別子を付して、レコード” 7 lamp#on” を構成する。また、変数情報取得手段 142 は、変数” lamp” の入力値” off” に変数識別子を付して、レコード” 8 lamp#off” を構成する。また、変数情報取得手段 142 は、変数” lamp” の出力値” on” に変数識別子を付して、レコード” 9 @lamp#on” を構成する。さらに、変数情報取得手段 142 は、変数” lamp” の出力値” off” に変数識別子を付して、レコード” 10 @lamp#off” を構成する。そして、変数情報取得手段 142 は、これらのレコードを変換表に追記する。

[0172] 次に、変数情報取得手段 142 は、図 14 の” Channel ch : m\_type;” を読み出す。そして、変数情報取得手段 142 は、予約語” Channel” から、この処理単位が変数の宣言である、と判断する。

[0173] 次に、変数情報取得手段 142 は、処理単位” Channel ch : m\_type;” から変数の型” m\_type” を取得する。次に、テスト元情報取得手段 141 は、この処理単位からテスト元情報を取得できない。

[0174] 次に、変数情報取得手段 142 は、型” m\_type” を用いて、変数” ch” の入力値 {m} を取得する。そして、変数情報取得手段 142 は、変数” ch” の入力値” m” に変数識別子を付して、レコード” 11 ch?m” を構成する。そして、変数情報取得手段 142 は、変換表にこのレコードを追記する。なお、予約語” Channel” はイベントを示し、入力値のみ存在し、出力値が存在しないことは、予め変数情報取得手段 142 は知っている。

[0175] 以上の処理により、図 15 に示す変数表が得られた。

[0176] 次に、テスト情報取得手段 243 は、図 14 の仕様記述情報を解釈し、以下のように命題論理式を取得する。

[0177] つまり、テスト情報取得手段 243 は、例えば、処理単位” Var op : onoff\_type {off};” を読み出し、当該処理単位から状態遷移前の変数に関する命題論理式「op#on||op#off」と「not(op#on)||not(op#off)」と、状態遷移後変数に関する命題論理式「@op#on||@op#off」と「not(@op#on)||not(@op#o

ff)」とを得る。

[0178] そして、テスト情報取得手段243は、図15の変数表を用いて、例えば、命題論理式「 $op\#on \mid op\#off$ 」に対して、変数識別子の集合「3 4 0」を得る。なお、「0」は命題論理式の終端を示す。また、テスト情報取得手段243は、図15の変数表を用いて、例えば、命題論理式「 $not(op\#on) \mid not(op\#off)$ 」に対して、変数識別子の集合「-3 -4 0」を得る。なお、「-」は「否定」を示す。つまり、「 $not(op\#on)$ 」は変数識別子「-3」となる。また、テスト情報取得手段243は、図15の変数表を用いて、命題論理式「 $@op\#on \mid @op\#off$ 」に対して、変数識別子の集合「5 6 0」を得る。さらに、テスト情報取得手段243は、図15の変数表を用いて、命題論理式「 $not(@op\#on) \mid not(@op\#off)$ 」に対して、変数識別子の集合「-5 -6 0」を得る。また、テスト情報取得手段243は、上記の命題論理式をコメント（行頭にcを付加）として付加する。そして、テスト情報取得手段243は、図16の(1)(2)(3)(4)を得る。

[0179] また、テスト情報取得手段243は、例えば、処理単位” $op==on \&\& lamp==off:ch?m: lamp==on;$ ”を読み出し、当該処理単位から命題論理式「 $op\#on \wedge lamp\#off \wedge ch?m \rightarrow @lamp\#on$ 」を得る。つまり、テスト情報取得手段143は、” $op==on$ ”を「 $op\#on$ 」、” $lamp==off$ ”を「 $lamp\#off$ 」、” $ch?m$ ”を「 $ch?m$ 」にし、” $\&\&$ ”を「 $\wedge$ 」に変換し、遷移前状態イベントとを「 $\wedge$ 」で連結する。そして、遷移後状態” $lamp==on$ ”を「 $@lamp\#on$ 」に変換し、「 $\rightarrow$ 」で連結する。以上の処理により、テスト情報取得手段243は、仕様記述情報から命題論理式を得る。

[0180] 次に、テスト情報取得手段243は、当該命題論理式「 $op\#on \wedge lamp\#off \wedge ch?m \rightarrow @lamp\#on$ 」を同値な論理式である「 $\neg op\#on \vee \neg lamp\#off \vee \neg ch?m \vee @lamp\#on$ 」に変形し、図15の変数表を用いて、変数識別子の集合に変換する処理を行う。そして、テスト情報取得手段243は、「9 -11 -3 -8 0」を得る。よって、テスト情報取得手段243は、図16の(5)の情報を得る。

[0181] テスト情報取得手段243は、以上のような処理を、すべての処理単位に対して行い、図16の命題論理式の集合を得る。つまり、実施の形態1で述べた処理系T1に対して、図14の仕様記述情報を与えれば、図16の命題論理式の集合が得られる。なお、図16の命題論理式の集合はSAT-solverへの入力である。

[0182] 次に、テスト情報取得手段243は、1以上の前提情報をも制約情報として用いて、すべての制約情報を満たすように、1以上の入力値および1以上の出力値の組であるテストスイートを1以上取得する。つまり、テスト情報取得手段243は、図14の仕様記述情報の制約「`op==on && lamp==off : ch?m : lamp==on;`」の前提情報「`op==on && lamp==off && ch?m`」を、仕様記述情報の制約情報として追加する。また、テスト情報取得手段243は、図15の変数表を参照して、「`op == on`」に対応する変数3、「`lamp == off`」に対応する変数8、「`ch?m`」に対応する変数11を拾い出し、図16に、(6)の3行を追加する。

また、状態遷移に関する制約が二つ以上存在するときには、前提情報のいずれかを満たすという制約を追加する。例えば「`op==on && lamp==off : ch?m : lamp==on;`」と「`op==on && lamp==on : null : true;`」が制約として存在する場合、テスト情報取得手段243は、前提情報「`op==on && lamp==off && ch?m`」と「`op==on && lamp==on && null`」とから制約「`(op==on && lamp==off && ch?m) || (op==on && lamp==on && null)`」を作成する。そして、テスト情報取得手段243は、前記制約を、当該制約と同値な論理式「`(op==on || op==on) && (op==on || lamp==on) && (op==on || null) && (lamp==off || op==on) && (lamp==off || lamp==on) && (lamp==off || null) && (ch?m || op==on) && (ch?m || lamp==on) && (ch?m || null)`」に変換したうえで図15の変数表を参照して、以下の9行の情報を得る。そして、テスト情報取得手段243は、9行をCNFファイルに追加する。

3 3 0

3 7 0

```

3   1   0
8   3   0
8   7   0
8   1   0
1 1   3   0
1 1   7   0
1 1   1   0

```

[0183] そして、ここで、テスト情報取得手段 243 が上記の 1 以上の前提情報をも制約情報として用いる処理を行わない場合、例えば、図 16 の命題論理式の集合から「 $op == off$  ,  $lamp == off$ ,  $ch?m$ ,  $@op==on$ ,  $@lamp == on$ 」というテストスイートが得られる。このテストスイートは、論理式の真偽値の定義上では、図 14 の仕様記述情報に記載されている制約「 $op==on \ \&\& \ lamp==off : ch?m : lamp==on;$ 」と「 $op==off \rightarrow lamp==off$ 」を満たすが、当該制約に現れる前提情報が有する遷移前条件「 $op == on \ \&\& \ lamp == off$ 」を満たしていない。そこで、テスト情報取得手段 243 は、図 16 の (6) を適用し、上記のテストスイートを解としない。

[0184] 以上のように、テスト情報取得手段 243 は、仕様記述情報から事前条件とイベント発生に相当する記述（前提情報）を抽出し、SAT-solver の入力形態に変換して制約を追加することを機械的に実行する。

[0185] （具体例 2）

具体例 2 は、イベント制約情報をも制約情報として用いて、無用なテストスイートの抑制を図る場合である。

[0186] 今、仕様記述情報格納部 12 は、図 14 に示す仕様記述情報を格納している。また、イベント制約情報格納手段 240 は「 $\neg null$ 」を格納している、とする。ここで、「 $\neg null$ 」は「イベントが  $null$  でない」ことを示す。つまり、テストの実施計画によっては、イベント発生を伴うテストスイートのみに注目したい場合がある。このとき、イベントを伴わないテストスイートを生成することは冗長である。そこで、イベント制約情報格納手段 240 に

「 $\neg$ null」を格納することによって、無用なテストスイートの発生を抑制する。

[0187] テスト情報取得手段243は、イベント制約情報格納手段240から「 $\neg$ null」を読み出す。そして、テスト情報取得手段243は、「 $\neg$ null」を制約として追加する。つまり、テスト情報取得手段243は、図15の変数表を参照し、「null#true」に対応する変数「1」を拾い出し、否定を表す「 $\neg$ 」の符号を付加して、図16の末尾に(7)の一行を追加する。

[0188] そして、テスト情報取得手段243が上記の制約情報として「 $\neg$ null」を用いる処理を行わない場合、例えば、図16の命題論理式の集合から「op == off , lamp == off, ch?m, @op==on, @lamp == on」というテストスイートが得られる。このテストスイートは、論理式の真偽値の定義上では、図14の仕様記述情報に記載されている制約「op==on && lamp==off : ch?m : lamp ==on;」と「op==off -> lamp==off」を満たすが、当該制約に現れる条件「 $\neg$ (null#true)」を満たしていない。そこで、テスト情報取得手段243は、図16の(7)を適用し、上記のテストスイートを解としない。

[0189] 以上より、テスト情報取得手段243は、図14の仕様を満たし、かつ「イベントが null でない」を満たすテストスイートを探索することとなる。

[0190] また、null 以外についても特定のイベント発生を伴うテストスイートの探索を同様の処理で抑制することができる。例えば、タイマのカウントアップを伴うテストスイートの出力を抑制する際には、テスト情報取得手段243は、「SubsystemCountUp」に対応する変数「2」を拾い出して、制約情報「-2 0」を追加すればよい。なお、かかる場合、イベント制約情報「 $\neg$ SubsystemCountUp」がイベント制約情報格納手段240に格納されている、とする。

[0191] (具体例3)

具体例3は、遷移前状態情報を削除することにより、無用なテストスイートの抑制を図る場合である。具体例3の動作を図18に示す。

[0192] 図16の命題論理式の集合(CNFファイル)から(6)(7)を除いた命題

論理式の集合を、実施の形態 1 で述べた T 2 で処理すると、DNF ファイルが出力される。つまり、従来技術におけるテスト情報取得手段は、図 16 の命題論理式の集合から (6) (7) を除いた命題論理式の集合から、図 17 に示すような値割り当てを取得する。なお、従来技術におけるテスト情報取得手段は、本実施の形態におけるテスト仕様生成装置 2 のテストスイート生成の防止機能を有さない。

[0193] 図 17 の一行目の解 (a) 「-1 -2 3 -4 5 -6 -7 8 9 -10 11」は、図 15 の変数表を参照すると、「 $\neg \text{null}\#true \wedge \neg \text{S ubsys temCountUp}\#true \wedge \text{op}\#on \wedge \neg \text{op}\#off \wedge @\text{op}\#on \wedge \neg @\text{op}\#off \wedge \neg \text{lamp}\#on \wedge \text{lamp}\#off \wedge @\text{lamp}\#on \wedge \neg @\text{lamp}\#off \wedge \text{ch?m}$ 」となる。つまり、図 17 の一行目の解のうち「-1, -2, 3, -4, -7, 8, 11」が遷移前状態情報と発生イベントの情報 (op==on かつ lamp==off かつ ch?m) である。また、「5, -6, 9, -10」が遷移後状態の情報 (@op==on かつ @lamp==on) である。

[0194] また、テスト情報取得手段 243 は、その論理否定「 $\text{null}\#true \vee \text{S ubsys temCountUp}\#true \vee \neg \text{op}\#on \vee \text{op}\#off \vee \neg @\text{op}\#on \vee @\text{op}\#off \vee \text{lamp}\#on \vee \neg \text{lamp}\#off \vee \neg @\text{lamp}\#on \vee @\text{lamp}\#off \vee \neg \text{ch?m}$ 」を符号化した数列「1 2 -3 4 -5 6 7 -8 -9 10 -11 0」を取得し、SAT-solver にフィードバックし、その結果、二つ目の解 (b) を計算する。ここで、テスト情報取得手段 243 は、フィードバックの際に、得られた最初の解の否定ではなく、遷移後状態「5, -6, 9, -10」 (@op==on かつ @lamp==on) の否定を取得する。つまり、テスト情報取得手段 243 は、テストスイートに対応する情報が有する遷移前状態とイベントの情報を削除して、かつ、当該制約情報が有する情報であり、削除した以外の部分の論理否定により構成された情報を取得する。具体的には、テスト情報取得手段 243 は、制約情報「(@op==on かつ @lamp==on) でないもの」を取得する。つまり、テスト情報取得手段 243 は、事後状態が (a) とは異なる解となるテストスイートを取得する。図 17 の (b) は事後状態に相当する部分が「5 -6 9 -10」で (a) と同一であり、制約情報「(@op==on かつ @lamp==on) でないもの」を満たさない。従って

、テスト情報取得手段243は、図17の(b)に対応するテストスイートを、最終的に出力しない。

[0195] 一方、図17の(c)は事後状態に相当する部分が「5 -6 -9 10」であり、制約情報「(@op==on かつ @lamp==on) でないもの」を満たす。従って、テスト情報取得手段243は、図17の(c)に対応するテストスイートを、最終的に出力する。以上より、テスト情報取得手段243は、結果として、事後状態に重複のないテストスイートの網羅的なリストを得ることができる。

[0196] (具体例4)

具体例4は、事後状態情報を削除することにより、無用なテストスイートの抑制を図る場合である。

[0197] 具体例3と同様に、図16の命題論理式の集合(CNFファイル)を、実施の形態1で述べたT2で処理すると、図17に示すような値割り当てが得られる。

[0198] また、図17の一行目の解(a)「-1 -2 3 -4 5 -6 -7 8 9 -10 11」は、図15の変数表を参照すると、「 $\neg \text{null\#true} \wedge \neg \text{S ubsys temCountU p\#true} \wedge \text{op\#on} \wedge \neg \text{op\#off} \wedge \text{@op\#on} \wedge \neg \text{@op\#off} \wedge \neg \text{lamp\#on} \wedge \text{lamp\#off} \wedge \text{@lamp\#on} \wedge \neg \text{@lamp\#off} \wedge \text{ch?m}$ 」となる。つまり「-1, -2, 3, -4, -7, 8, 11」が事前状態情報と発生イベントの情報(op==on かつ lamp==off かつ ch?m)である。

[0199] そして、テスト情報取得手段243は、事前状態情報と発生イベントの論理否定「(op==on かつ lamp==off かつ ch?m)の否定」を取得する。そして、テスト情報取得手段243は、元の制約をすべて満たし、かつ「(op==on かつ lamp==off かつ ch?m) でないもの」を満たすテストスイートを取得する。つまり、図17の(b)が「(op==on かつ lamp==off かつ ch?m) でないもの」を満たすとして、テスト情報取得手段243は、図17の(b)をテストスイートとして採用する。以上により、テスト情報取得手段243は、結果として、事前状態に重複のないテストスイートの網羅的なリストを得る。

。

[0200] 以上、本実施の形態によれば、不適切なテストスイートの生成を防止できる。

[0201] さらに、本実施の形態におけるテスト仕様生成装置を実現するソフトウェアは、以下のようなプログラムである。つまり、このプログラムは、記憶媒体に、テストで利用する情報であるテスト情報と、対象の仕様について記述する言語である仕様記述言語で記述された前記対象の仕様に関する情報であり、1以上の変数の宣言を含む情報である仕様記述情報を格納しており、コンピュータを、前記変数の宣言を用いて、1以上の各変数の入力値を少なくとも有する1以上のテスト情報を取得するテスト情報取得部と、前記テスト情報取得部が取得した1以上のテスト情報を前記テスト情報格納部に蓄積するテスト情報蓄積部として機能させるためのプログラム、である。

[0202] また、上記プログラムにおいて、前記仕様記述情報は、1以上の変数の宣言、および制約に関する記述である1以上の制約情報を有し、前記テスト情報取得手段は、前記変数情報取得手段が取得した変数と当該変数の型から入力値および出力値を変数ごとに取得し、かつ、前記すべての制約情報を満たすように、1以上の入力値および1以上の出力値の組であるテストスイートを1以上取得するプログラムである。

[0203] また、上記プログラムにおいて、前記制約情報は、1以上の変数の値の集合を有する状態である遷移前状態と当該遷移前状態に対して発生するイベントと前記状態の場合に前記イベントが発生した場合に移行する状態である遷移後状態とを有する状態遷移への制約情報であることは好適である。

[0204] また、上記プログラムにおいて、前記テスト情報取得手段は、前記1以上の制約情報が有する遷移前状態とイベントとの組である1以上の前提情報をも制約情報として用いて、すべての制約情報を満たすように、1以上の入力値および1以上の出力値の組であるテストスイートを1以上取得することは好適である。

[0205] また、上記プログラムにおいて、前記テスト情報取得部は、イベントに関

する制約を示す 1 以上のイベント制約情報を格納し得るイベント制約情報格納手段をさらに具備し、前記テスト情報取得手段は、前記 1 以上のイベント制約情報をも制約情報として用いて、すべての制約情報を満たすように、1 以上の入力値および 1 以上の出力値の組であるテストスイートを 1 以上取得することは好適である。

[0206] また、上記プログラムにおいて、前記テスト情報取得手段は、既に取得したテストスイートに対応する制約情報から、当該制約情報が有する遷移前状態とイベントに関する情報を削除し、かつ、当該遷移前状態以外の当該制約情報が有する部分の論理否定を構成した 1 以上の第二制約情報を取得し、当該 1 以上の第二制約情報をも制約情報として用いて、すべての制約情報を満たすように、1 以上の入力値および 1 以上の出力値の組であるテストスイートを 1 以上取得することは好適である。

[0207] また、上記プログラムにおいて、前記テスト情報取得手段は、前記 1 以上の各制約情報から、各制約情報が有する遷移後状態を削除し、かつ、前記各制約情報が有する部分であり、当該遷移後状態以外の部分の論理否定を構成した 1 以上の第二制約情報を取得し、当該 1 以上の第二制約情報をも制約情報として用いて、すべての制約情報を満たすように、1 以上の入力値および 1 以上の出力値の組であるテストスイートを 1 以上取得することは好適である。

[0208] また、図 19 は、本明細書で述べたプログラムを実行して、上述した実施の形態のテスト仕様生成装置等を実現するコンピュータの外観を示す。上述の実施の形態は、コンピュータハードウェア及びその上で実行されるコンピュータプログラムで実現され得る。図 19 は、このコンピュータシステム 340 の概観図であり、図 20 は、コンピュータシステム 340 の内部構成を示す図である。

[0209] 図 19 において、コンピュータシステム 340 は、FD ドライブ 341 1、CD-ROM ドライブ 341 2 を含むコンピュータ 341 と、キーボード 342 と、マウス 343 と、モニタ 344 とを含む。

- [0210] 図20において、コンピュータ341は、FDドライブ3411、CD-ROMドライブ3412に加えて、MPU3413と、CD-ROMドライブ3412及びFDドライブ3411に接続されたバス3414と、ブートアッププログラム等のプログラムを記憶するためのROM3415とに接続され、アプリケーションプログラムの命令を一時的に記憶するとともに一時記憶空間を提供するためのRAM3416と、アプリケーションプログラム、システムプログラム、及びデータを記憶するためのハードディスク3417とを含む。ここでは、図示しないが、コンピュータ341は、さらに、LANへの接続を提供するネットワークカードを含んでも良い。
- [0211] コンピュータシステム340に、上述した実施の形態のテスト仕様生成装置等の機能を実行させるプログラムは、CD-ROM3501、またはFD3502に記憶されて、CD-ROMドライブ3412またはFDドライブ3411に挿入され、さらにハードディスク3417に転送されても良い。これに代えて、プログラムは、図示しないネットワークを介してコンピュータ341に送信され、ハードディスク3417に記憶されても良い。プログラムは実行の際にRAM3416にロードされる。プログラムは、CD-ROM3501、FD3502またはネットワークから直接、ロードされても良い。
- [0212] プログラムは、コンピュータ341に、上述した実施の形態のテスト仕様生成装置等の機能を実行させるオペレーティングシステム(OS)、またはサードパーティープログラム等は、必ずしも含まなくても良い。プログラムは、制御された態様で適切な機能(モジュール)を呼び出し、所望の結果が得られるようにする命令の部分のみを含んでいれば良い。コンピュータシステム340がどのように動作するかは周知であり、詳細な説明は省略する。
- [0213] また、上記プログラムを実行するコンピュータは、単数であってもよく、複数であってもよい。すなわち、集中処理を行ってもよく、あるいは分散処理を行ってもよい。
- [0214] また、上記各実施の形態において、各処理(各機能)は、単一の装置(シ

ステム)によって集中処理されることによって実現されてもよく、あるいは、複数の装置によって分散処理されることによって実現されてもよい。

[0215] 本発明は、以上の実施の形態に限定されることなく、種々の変更が可能であり、それらも本発明の範囲内に包含されるものであることは言うまでもない。

### 産業上の利用可能性

[0216] 以上のように、本発明にかかるテスト仕様生成装置は、対象の設計時にテスト設計のための情報を記載すれば適切なテスト仕様に関する情報が自動生成できる、という効果を有し、テスト支援装置等として有用である。

### 符号の説明

- [0217] 1、2 テスト仕様生成装置
- 1 1 テスト情報格納部
  - 1 2 仕様記述情報格納部
  - 1 3 受付部
  - 1 4、2 4 テスト情報取得部
  - 1 5 テスト情報蓄積部
  - 1 6 テスト情報出力部
  - 1 4 1 テスト元情報取得手段
  - 1 4 1 テスト情報元取得手段
  - 1 4 2 変数情報取得手段
  - 1 4 3、2 4 3 テスト情報取得手段

## 請求の範囲

- [請求項1]      テストで利用する情報であるテスト情報を格納し得るテスト情報格納部と、
- 対象の仕様について記述する言語である仕様記述言語で記述された前記対象の仕様に関する情報であり、1 以上の変数の宣言を含む情報であり、テスト情報を生成するためのテスト元情報を含む情報である仕様記述情報を格納し得る仕様記述情報格納部と、
- 前記変数の宣言および前記テスト元情報を用いて、1 以上の各変数の入力値を少なくとも有する1 以上のテスト情報を取得するテスト情報取得部と、
- 前記テスト情報取得部が取得した1 以上のテスト情報を前記テスト情報格納部に蓄積するテスト情報蓄積部とを具備するテスト仕様生成装置。
- [請求項2]      前記変数は、型を有し、
- 前記仕様記述情報は、
- 前記1 以上の変数のうち、1 以上の各変数に対応付けて、各変数が採り得る値の中で、テストで使用する値を特定する情報であるテスト元情報を有し、
- 前記テスト情報取得部は、
- 前記仕様記述情報から、前記テスト元情報を前記変数と対応付けて取得するテスト元情報取得手段と、
- 前記仕様記述情報から、テスト元情報を有さない変数と当該変数の型を取得する変数情報取得手段と、
- 前記テスト情報取得手段が取得したテスト元情報から、各テスト元情報に対応する変数の1 以上の入力値を変数ごとに取得し、かつ、前記変数情報取得手段が取得した変数と当該変数の型から入力値を変数ごとに取得するテスト情報取得手段とを具備する請求項1 記載のテスト仕様生成装置。

- [請求項3] 前記テスト情報取得手段は、  
前記テスト元情報取得手段が取得したテスト元情報から、各テスト元情報に対応する変数の1以上の入力値および出力値を変数ごとに取得し、かつ、前記変数情報取得手段が取得した変数と当該変数の型から入力値および出力値を変数ごとに取得する請求項2記載のテスト仕様生成装置。
- [請求項4] 前記仕様記述情報は、  
1以上の変数の宣言、および制約に関する記述である1以上の制約情報を有し、  
前記テスト情報取得手段は、  
前記テスト元情報取得手段が取得したテスト元情報から、各テスト元情報に対応する変数の1以上の入力値および出力値を変数ごとに取得し、かつ、前記変数情報取得手段が取得した変数と当該変数の型から入力値および出力値を変数ごとに取得し、かつ、前記すべての制約情報を満たすように、1以上の入力値および1以上の出力値の組であるテストスイートを1以上取得する請求項3記載のテスト仕様生成装置。
- [請求項5] 前記制約情報は、1以上の変数の値の集合を有する状態である遷移前状態と当該遷移前状態に対して発生するイベントと前記状態の場合に前記イベントが発生した場合に移行する状態である遷移後状態と明示的にイベントを発生させるアクションとを有する状態遷移情報、または1以上の変数の値の集合を有する第一状態と当該第一状態の場合に必ず採用される1以上の変数の値の集合を有する第二状態とを有する不変条件情報である請求項4記載のテスト仕様生成装置。
- [請求項6] 前記テスト情報取得手段は、  
前記1以上の制約情報が有する遷移前状態とイベントとの組である1以上の前提情報をも制約情報として用いて、すべての制約情報を満たすように、1以上の入力値および1以上の出力値の組であるテストス

イートを1以上取得する請求項5記載のテスト仕様生成装置。

[請求項7]

前記テスト情報取得部は、

イベントに関する制約を示す1以上のイベント制約情報を格納し得るイベント制約情報格納手段をさらに具備し、

前記テスト情報取得手段は、

前記1以上のイベント制約情報をも制約情報として用いて、すべての制約情報を満たすように、1以上の入力値および1以上の出力値の組であるテストスイートを1以上取得する請求項5記載のテスト仕様生成装置。

[請求項8]

前記テスト情報取得手段は、

既に取得したテストスイートに対応する制約情報から、当該制約情報が有する遷移前状態とイベントに関する情報を削除し、かつ、当該遷移前状態以外の当該制約情報が有する部分の論理否定を構成した1以上の第二制約情報を取得し、当該1以上の第二制約情報をも制約情報として用いて、すべての制約情報を満たすように、1以上の入力値および1以上の出力値の組であるテストスイートを1以上取得する請求項5記載のテスト仕様生成装置。

[請求項9]

前記テスト情報取得手段は、

前記1以上の各制約情報から、各制約情報が有する遷移後状態を削除し、かつ、前記各制約情報が有する部分であり、当該遷移後状態以外の部分の論理否定を構成した1以上の第二制約情報を取得し、当該1以上の第二制約情報をも制約情報として用いて、すべての制約情報を満たすように、1以上の入力値および1以上の出力値の組であるテストスイートを1以上取得する請求項5記載のテスト仕様生成装置。

[請求項10]

記憶媒体に、

テストで利用する情報であるテスト情報と、

対象の仕様について記述する言語である仕様記述言語で記述された前記対象の仕様に関する情報であり、1以上の変数の宣言を含む情報で

あり、テスト情報を生成するためのテスト元情報を含む情報である仕様記述情報を格納しており、

テスト情報取得部、およびテスト情報蓄積部により実現され得るテスト仕様生成方法であって、

前記テスト情報取得部が、前記変数の宣言および前記テスト元情報を用いて、1以上の各変数の入力値を少なくとも有する1以上のテスト情報を取得するテスト情報取得ステップと、

前記テスト情報蓄積部が、前記テスト情報取得ステップで取得された1以上のテスト情報を前記テスト情報格納部に蓄積するテスト情報蓄積ステップとを具備するテスト仕様生成方法。

[請求項11]

記憶媒体に、

テストで利用する情報であるテスト情報と、

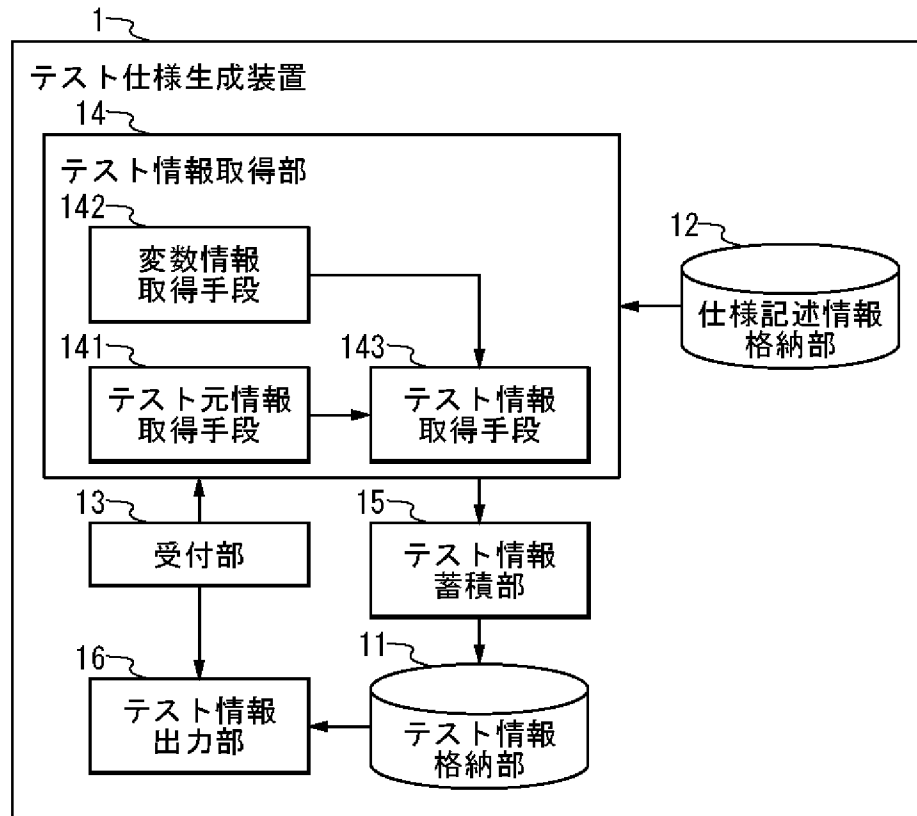
対象の仕様について記述する言語である仕様記述言語で記述された前記対象の仕様に関する情報であり、1以上の変数の宣言を含む情報であり、テスト情報を生成するためのテスト元情報を含む情報である仕様記述情報を格納しており、

コンピュータを、

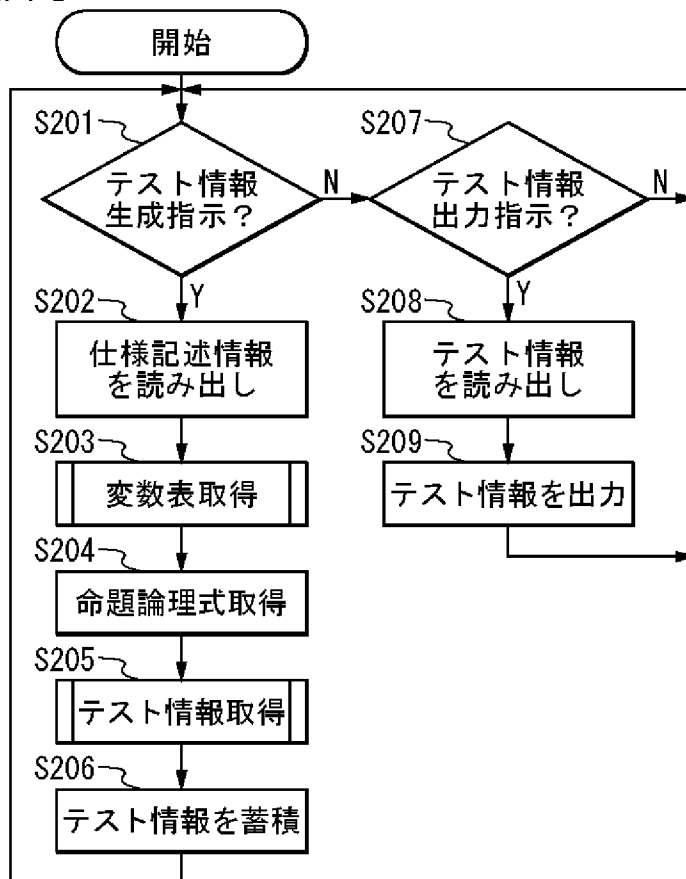
前記変数の宣言および前記テスト元情報を用いて、1以上の各変数の入力値を少なくとも有する1以上のテスト情報を取得するテスト情報取得部と、

前記テスト情報取得部が取得した1以上のテスト情報を前記テスト情報格納部に蓄積するテスト情報蓄積部として機能させるためのプログラム。

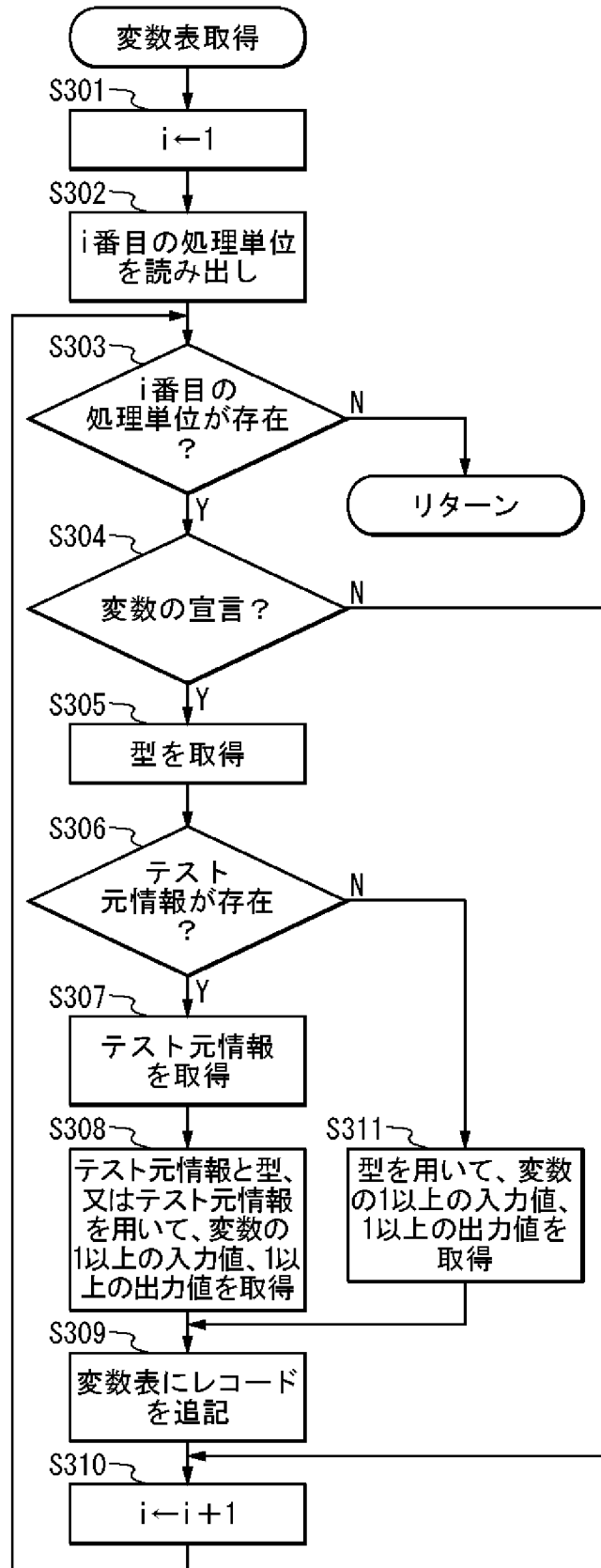
[図1]



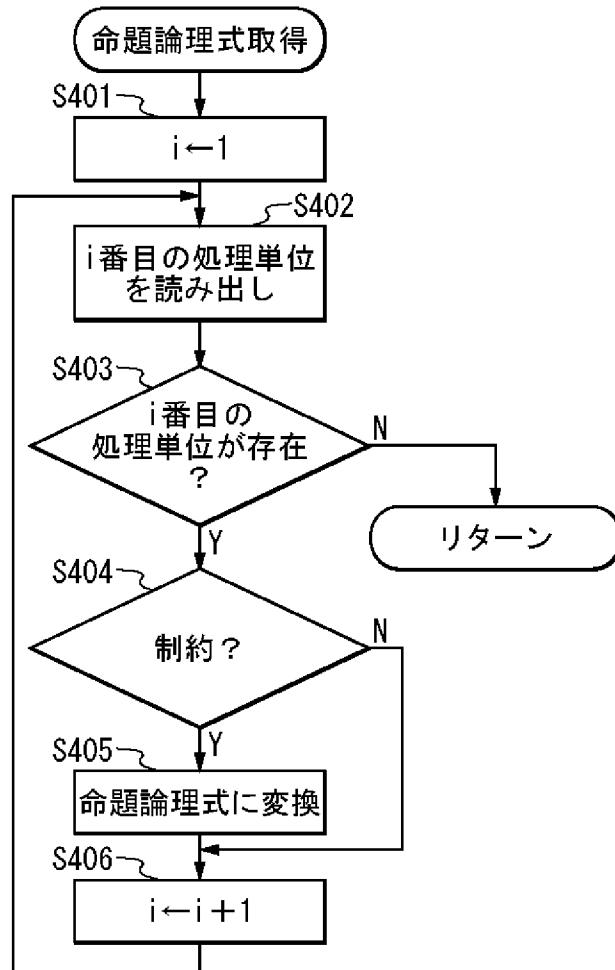
[図2]



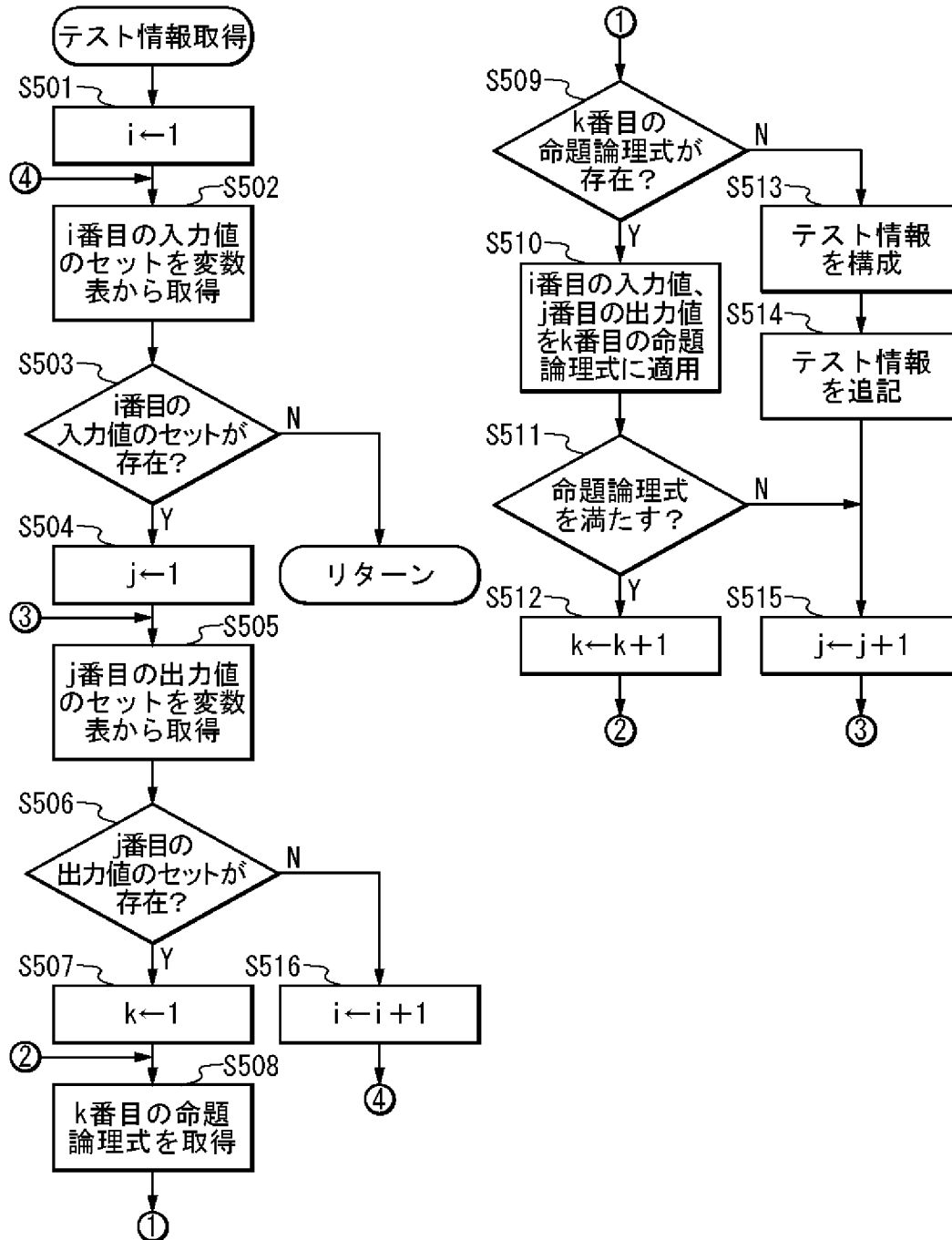
[図3]



[図4]



[図5]



[図6]

|                   |         |                                         |     |
|-------------------|---------|-----------------------------------------|-----|
| Extern            |         |                                         |     |
|                   | Data    | onoff_type; // from main                | (1) |
| Class B           |         |                                         |     |
|                   | Data    | m_type {m};                             | (2) |
|                   | Var     | op : onoff_type {off} testedwith {off}; | (3) |
|                   |         | lamp : onoff_type {off};                | (4) |
|                   | Channel | ch : m_type;                            | (5) |
| Function Lighting |         |                                         |     |
| Require           |         | op==on && lamp==off : ch?m : lamp==on;  | (6) |
|                   |         | op==off -> lamp==off;                   | (7) |

[図7]

```

c --- CNFNum to VarAssign table ---
c      1      null#true
c      2      SubsystemCountUp#true
c      3      op#off
c      4      @op#off
c      5      lamp#on
c      6      lamp#off
c      7      @lamp#on
c      8      @lamp#off
c      9      ch?m
c --- end ---

```

[図8]

```

c --- cnf file ---
c --- Var : op (pre cond) ---
c op#off _____ (1)
3 0 _____ (2)
c --- Var : op (post cond) ---
c @op#off _____ (3)
4 0 _____ (4)
c --- CountUp Invariant ( op ) ---
c SubsystemCountUp#true && op#off -> @op#off
-2 -3 4 0
c --- Var : lamp (pre cond) ---
c lamp#on || lamp#off
5 6 0
c not(lamp#on) || not(lamp#off)
-5 -6 0
c --- Var : lamp (post cond) ---
c @lamp#on || @lamp#off
7 8 0
c not(@lamp#on) || not(@lamp#off)
-7 -8 0
c --- CountUp Invariant ( lamp ) ---
c SubsystemCountUp#true && lamp#on -> @lamp#on
-2 -5 7 0
c SubsystemCountUp#true && lamp#off -> @lamp#off
-2 -6 8 0
c >>>>>>>> Function : Lighting (begin) >>>>>>>>
c op == on && lamp == off : ch?m : lamp == on _____ (5)
c op == off -> lamp == off (this state) _____ (6)
-3 6 0 _____ (7)
c op == off -> lamp == off (next state) _____ (8)
-4 8 0 _____ (9)
c <<<<<<<<< Lighting (end) <<<<<<<<<
c --- all event [null#true, SubsystemCountUp#true, ch?m] ---
1 2 9 0
c event exclusive
-1 -2 0
-1 -9 0
-2 -9 0

```

[図9]

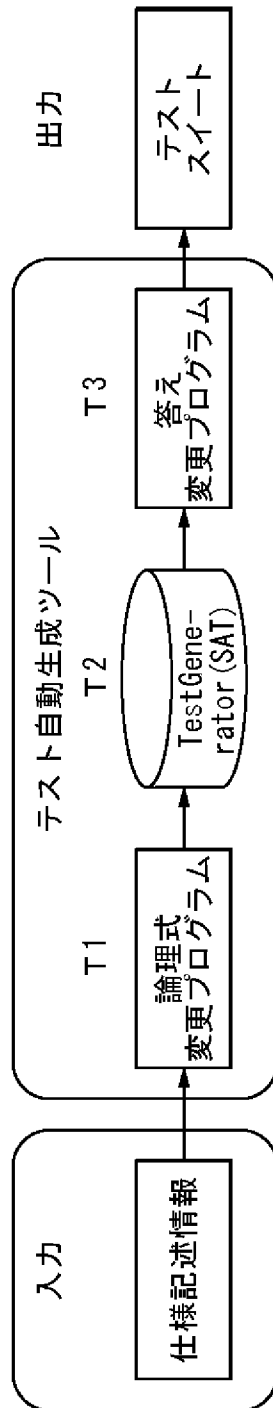
|                        |     |
|------------------------|-----|
| -1 2 3 4 -5 6 -7 8 -9  | (1) |
| -1 -2 3 4 -5 6 -7 8 9  | (2) |
| 1 -2 3 4 -5 6 -7 8 -9  | (3) |
| UNSAT                  |     |
| ---                    |     |
| finally result Num : 3 |     |
| ---                    |     |

[図10]

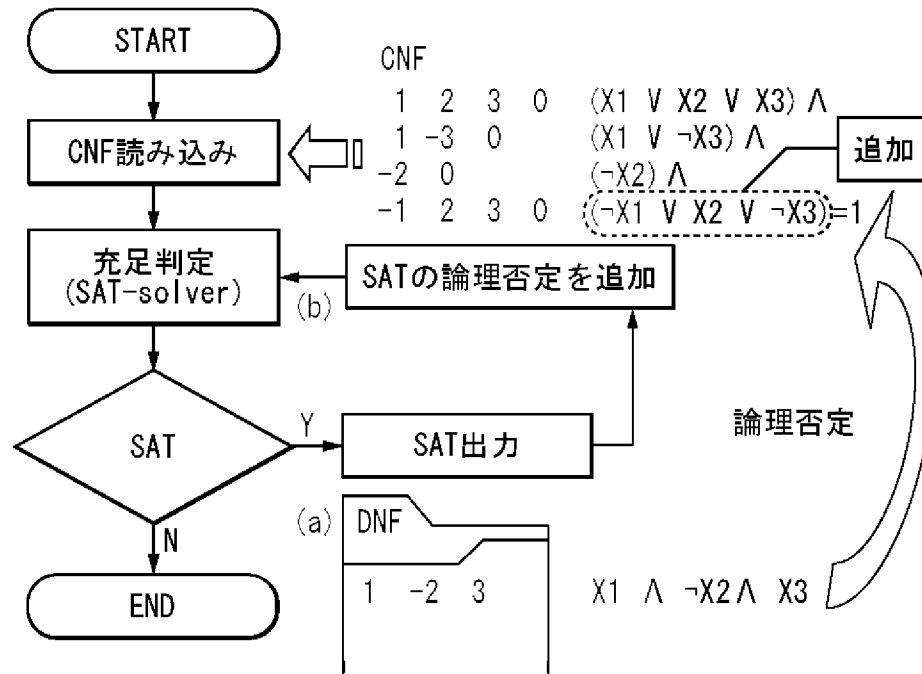
| 入力値 |      |                  | 出力値 |       |        |
|-----|------|------------------|-----|-------|--------|
| op  | lamp | event            | @op | @lamp | action |
| off | off  | SubsystemCountUp | off | off   | null   |
| off | off  | ch?m             | off | off   | null   |
| off | off  | null             | off | off   | null   |

テストスイート

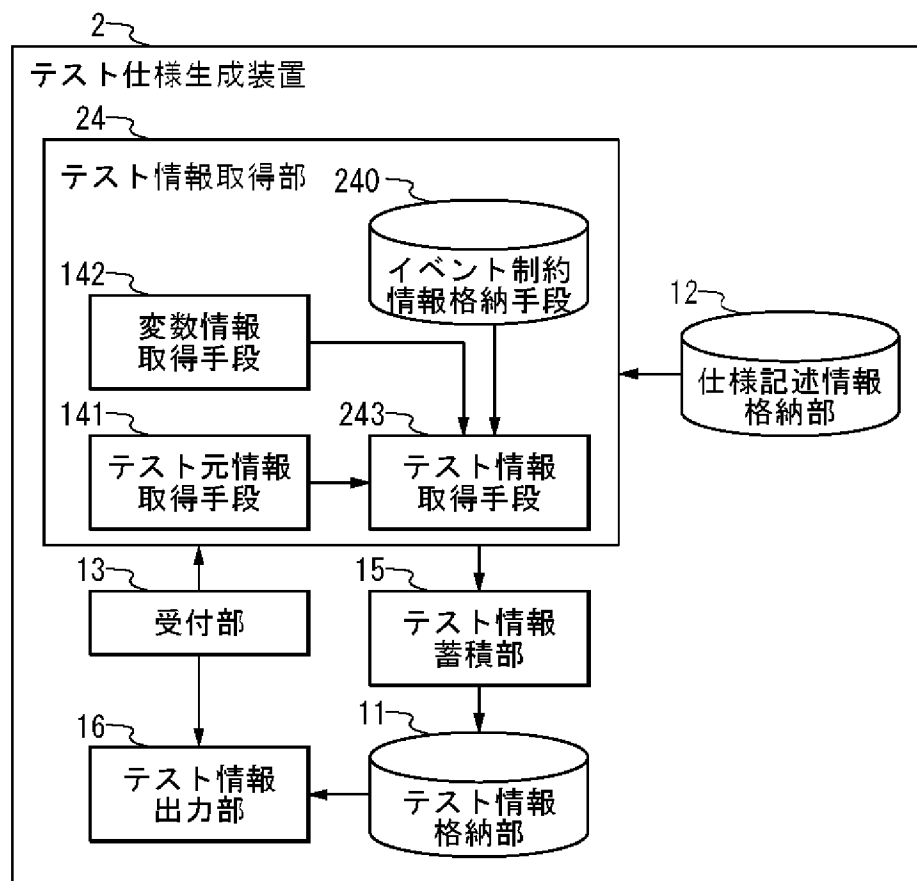
[図11]



[図12]



[図13]



[図14]

|                          |                |                                       |     |
|--------------------------|----------------|---------------------------------------|-----|
| <b>Extern</b>            |                |                                       |     |
|                          | <b>Data</b>    | onoff_type; // from main              | (1) |
| <b>Class B</b>           |                |                                       |     |
|                          | <b>Data</b>    | m_type {m};                           | (2) |
|                          | <b>Var</b>     | op : onoff_type {off}                 | (3) |
|                          |                | lamp : onoff_type {off};              | (4) |
|                          | <b>Channel</b> | ch : m_type;                          | (5) |
| <b>Function Lighting</b> |                |                                       |     |
|                          | <b>Require</b> |                                       |     |
|                          |                | op==on && lamp==off: ch?m : lamp==on; | (6) |
|                          |                | op==off-> lamp==off;                  | (7) |

[図15]

|   |     |                           |     |
|---|-----|---------------------------|-----|
| c | --- | CNFNum to VarAssign table | --- |
| c | 1   | null#true                 |     |
| c | 2   | SubsystemCountUp#true     |     |
| c | 3   | op#on                     |     |
| c | 4   | op#off                    |     |
| c | 5   | @ op#on                   |     |
| c | 6   | @ op#off                  |     |
| c | 7   | lamp#on                   |     |
| c | 8   | lamp#off                  |     |
| c | 9   | @ lamp#on                 |     |
| c | 10  | @ lamp#off                |     |
| c | 11  | ch? m                     |     |
| c | --- | end                       | --- |

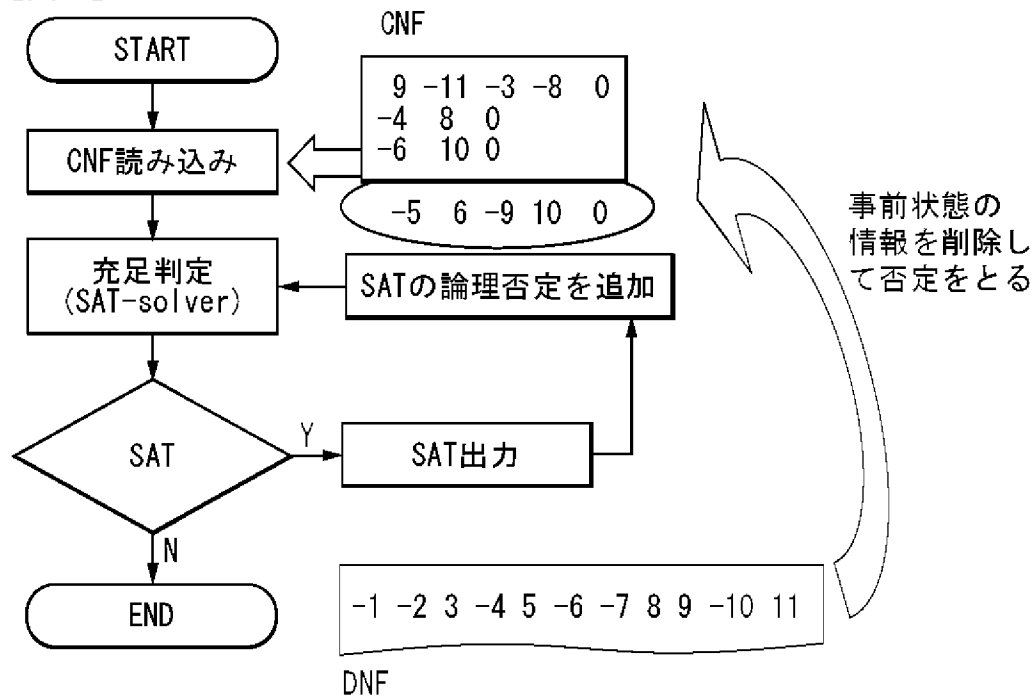
[図16]

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>c --- cnf file ---</pre><br><pre>c --- Var : op (pre cond) ---</pre><br><pre>c op#on    op#off</pre><br><div>(1)</div> <hr/> <pre>c not(op#on)    not(op#off)</pre><br><div>(2)</div> <hr/> <pre>c --- Var : op (post cond) ---</pre><br><pre>c @ op#on    @ op#off</pre><br><div>(3)</div> <hr/> <pre>c not( @ op#on )    not( @ op#off )</pre><br><div>(4)</div> <hr/> <pre>c --- CountUp Invariant ( op )---</pre><br><pre>c SubsystemCountUp#true &amp;&amp; op#on -&gt; @ op#on</pre><br><pre>-2 -3 5 0</pre><br><pre>c SubsystemCountUp#true &amp;&amp; op#off -&gt; @ op#off</pre><br><pre>-2 -4 6 0</pre> | <pre>c --- CountUp Invariant ( lamp )---</pre><br><pre>c SubsystemCountUp#true &amp;&amp; lamp#on -&gt; @ lamp#on</pre><br><pre>-2 -7 9 0</pre><br><pre>c SubsystemCountUp#true &amp;&amp; lamp#off -&gt; @ lamp#off</pre><br><pre>-2 -8 10 0</pre><br><br><pre>c &gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt; Function : Lighting (begin) &gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;</pre><br><pre>c op == on &amp;&amp; lamp == off : ch?m : lamp == on</pre><br><div>(5)</div> <hr/> <pre>c op == off -&gt; lamp == off (this state)</pre><br><pre>-4 8 0</pre><br><br><pre>c op == off -&gt; lamp == off (next state)</pre><br><pre>-6 10 0</pre><br><br><pre>c &lt;&lt;&lt;&lt;&lt;&lt;&lt;&lt;&lt;&lt; Lighting (end) &lt;&lt;&lt;&lt;&lt;&lt;&lt;&lt;&lt;&lt;</pre><br><br><pre>c --- all event [null#true, SubsystemCountUp#true, ch? m] ---</pre><br><pre>1 2 11 0</pre><br><br><pre>c event exclusive</pre><br><pre>-1 -2 0</pre><br><pre>-1 -11 0</pre><br><pre>-2 -11 0</pre><br><br><div>(6)</div> <hr/> <pre>3 0 }<br/>8 0 }<br/>11 0 }</pre><br><br><div>(7)</div> <hr/> <pre>-1 0</pre> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

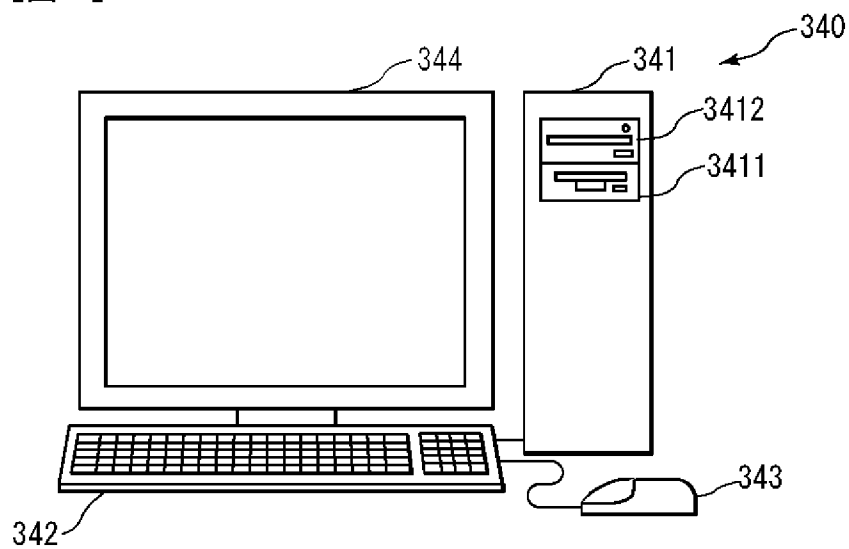
[図17]

|       |    |   |    |   |    |    |   |    |     |     |     |
|-------|----|---|----|---|----|----|---|----|-----|-----|-----|
| -1    | -2 | 3 | -4 | 5 | -6 | -7 | 8 | 9  | -10 | 11  | (a) |
| 1     | -2 | 3 | -4 | 5 | -6 | -7 | 8 | 9  | -10 | -11 | (b) |
| 1     | -2 | 3 | -4 | 5 | -6 | -7 | 8 | -9 | 10  | -11 | (c) |
| [以下略] |    |   |    |   |    |    |   |    |     |     |     |

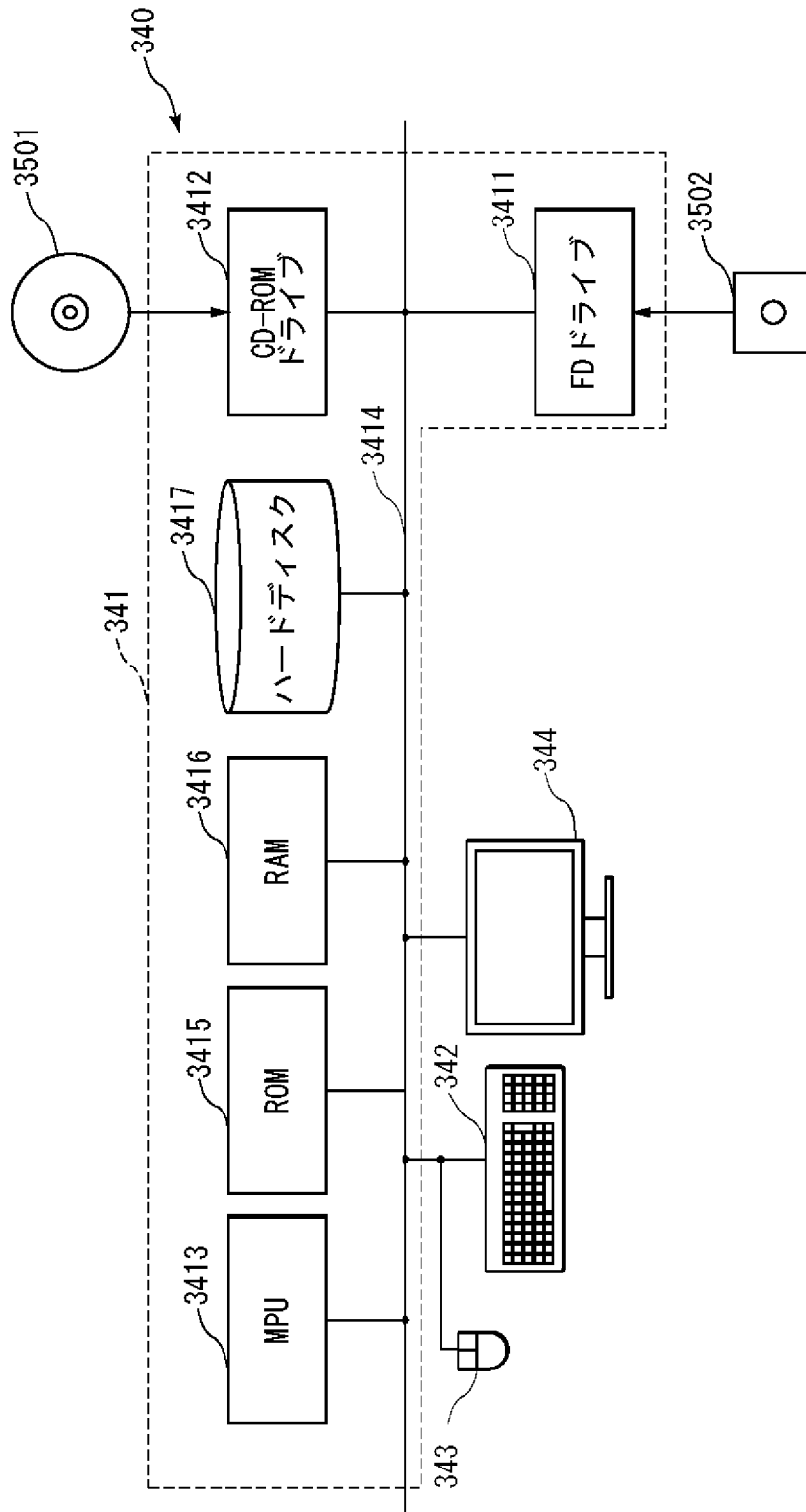
[図18]



[図19]



[図20]



# INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2012/056368

## A. CLASSIFICATION OF SUBJECT MATTER

G06F11/36(2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F11/36, G06F11/28, G06F9/44

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

|                           |           |                            |           |
|---------------------------|-----------|----------------------------|-----------|
| Jitsuyo Shinan Koho       | 1922-1996 | Jitsuyo Shinan Toroku Koho | 1996-2012 |
| Kokai Jitsuyo Shinan Koho | 1971-2012 | Toroku Jitsuyo Shinan Koho | 1994-2012 |

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

JSTPlus (JDreamII)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages                                                                                   | Relevant to claim No. |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| X<br>A    | JP 2008-293382 A (Fuji Electric Holdings Co., Ltd.),<br>04 December 2008 (04.12.2008),<br>paragraphs [0016] to [0031], [0048] to [0057],<br>[0087]<br>(Family: none) | 1-4, 10, 11<br>5-9    |
| A         | JP 2007-323573 A (International Business Machines Corp.),<br>13 December 2007 (13.12.2007),<br>paragraphs [0024] to [0054]<br>& US 2008/0072100 A1                   | 1-11                  |

☒ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

\* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search  
10 April, 2012 (10.04.12)

Date of mailing of the international search report  
17 April, 2012 (17.04.12)

Name and mailing address of the ISA/  
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

# INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2012/056368

## C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages                                                                                                                                                                                                                                                                                                                                       | Relevant to claim No. |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| A         | Takahiro SEINO et al., "Generating test cases for implementing concurrent systems based on the OTS/CafeOBJ method", IEICE Technical Report, 26 July 2010 (26.07.2010), vol.110, no. 161, pages 7 to 12, particularly, descriptions in passages 3 to 5                                                                                                                                                                    | 1-11                  |
| A         | JP 8-55041 A (Matsushita Electric Industrial Co., Ltd.),<br>27 February 1996 (27.02.1996),<br>paragraphs [0019] to [0030]<br>(Family: none)                                                                                                                                                                                                                                                                              | 1-11                  |
| P,A       | Eun Hye CHOI et al., SENS - Specification Language for Embedded Network Systems - toward Automatic Test Generation, Programming Science Technical Report, [online], 2011.8, Collaborative Research Team for Verification and Specificaton, National Institute of Advanced Industrial Science and Technology, [retrieved on 09 April 2012 (09.04.2012)], Internet <URL:http://cfv.jp/cvs/introduction/pdf/PS2011-002.pdf> | 1-11                  |

## A. 発明の属する分野の分類 (国際特許分類 (I P C))

Int.Cl. G06F11/36(2006.01)i

## B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (I P C))

Int.Cl. G06F11/36, G06F11/28, G06F9/44

最小限資料以外の資料で調査を行った分野に含まれるもの

|             |                     |
|-------------|---------------------|
| 日本国実用新案公報   | 1 9 2 2 - 1 9 9 6 年 |
| 日本国公開実用新案公報 | 1 9 7 1 - 2 0 1 2 年 |
| 日本国実用新案登録公報 | 1 9 9 6 - 2 0 1 2 年 |
| 日本国登録実用新案公報 | 1 9 9 4 - 2 0 1 2 年 |

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

JSTPlus (JDreamII)

## C. 関連すると認められる文献

| 引用文献の<br>カテゴリー* | 引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示                                                                       | 関連する<br>請求項の番号     |
|-----------------|---------------------------------------------------------------------------------------------------------|--------------------|
| X<br>A          | JP 2008-293382 A (富士電機ホールディングス株式会社)<br>2008.12.04, 段落【0016】 - 【0031】, 【0048】 - 【0057】, 【0087】 (ファミリーなし) | 1-4, 10, 11<br>5-9 |
| A               | JP 2007-323573 A (インターナショナル・ビジネス・マシーンズ・コーポレーション) 2007.12.13, 段落【0024】 - 【0054】<br>& US 2008/0072100 A1  | 1-11               |

☒ C 欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

## \* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの  
「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの  
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)  
「O」口頭による開示、使用、展示等に言及する文献  
「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの  
「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの  
「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの  
「&」同一パテントファミリー文献

国際調査を完了した日

1 0 . 0 4 . 2 0 1 2

国際調査報告の発送日

1 7 . 0 4 . 2 0 1 2

国際調査機関の名称及びあて先

日本国特許庁 (I S A / J P)  
郵便番号 1 0 0 - 8 9 1 5  
東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

稲垣 良一

5 B

4 0 5 5

電話番号 0 3 - 3 5 8 1 - 1 1 0 1 内線 3 5 4 4

| C (続き) . 関連すると認められる文献 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| 引用文献の<br>カテゴリー*       | 引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示                                                                                                                                                                                                                                                                                                                                                                                                                              | 関連する<br>請求項の番号 |
| A                     | 清野 貴博 他, O T S / C a f e O B J 法に基づく並行システムの実装とテスト生成, 電子情報通信学会技術研究報告, 2010.07.26, Vol.110, No.161, pp.7-12<br>特に、第3節－第5節の記載を参照。                                                                                                                                                                                                                                                                                                                                 | 1-11           |
| A                     | JP 8-55041 A (松下電器産業株式会社) 1996.02.27,<br>段落【0019】－【0030】 (ファミリーなし)                                                                                                                                                                                                                                                                                                                                                                                             | 1-11           |
| P, A                  | Eun Hye CHOI et al., SENS - Specification Language for Embedded Network Systems - toward Automatic Test Generation, Programming Science Technical Report, [online], 2011.8, Collaborative Research Team for Verification and Specifcation, National Institute of Advanced Industrial Science and Technology, [平成24年 4月 9日検索], インターネット<URL: <a href="http://cfv.jp/cvs/introduction/pdf/PS2011-002.pdf">http://cfv.jp/cvs/introduction/pdf/PS2011-002.pdf</a> > | 1-11           |